

WebSpeed Installation and Configuration Guide

© 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: 4602
Item Number: 89437;V31D

Contents

Preface xiii
Purpose xiii
Audience xiii
Organization Of This Manual xiii
Windows and UNIX Path Conventions xv
Typographical Conventions xvi
Syntax Notation xvii
WebSpeed Messages xxi
Other Useful Documentation xxii
Getting Started xxii
Development Tools xxiii
Reporting Tools xxv
4GL xxv
Database xxvi
DataServers xxvii
SQL-89/Open Access xxvii
SQL-92 xxviii
Deployment xxix
WebSpeed xxix
Reference xxx

1. Installation Requirements 1–1
1.1 Quick Tour Of the WebSpeed Components 1–2
1.2 System Requirements 1–2
1.2.1 Hardware and Networking Requirements 1–3
1.2.2 Java Requirements 1–4
1.2.3 WebSpeed Component Requirements On Windows 1–6
1.2.4 WebSpeed Component Requirements On UNIX 1–9

1.3 WebSpeed Agent/Database Server Compatibility 1–10
1.4 WebSpeed Licensing Information 1–10
1.5 Product Life Cycle 1–12

2. Overview Of the WebSpeed Architecture 2–1
2.1 WebSpeed Components 2–2
2.2 The Progress Administration Framework 2–5
2.2.1 AdminServer 2–5
2.2.2 Progress Explorer 2–6
2.2.3 Command-line Utilities 2–6
2.3 The ubroker.properties File 2–7
2.4 How WebSpeed Components Work Together 2–8

3. WebSpeed Configurations 3–1
3.1 Configuration Choices 3–2
3.1.1 Installing the Progress Explorer 3–2
3.1.2 WebSpeed WorkShop Or ProVision Plus 3–2
3.1.3 Deciding Which Installation Level To Use 3–3
3.1.4 Eliminating NameServer Usage 3–4
3.2 Single-machine Configuration 3–5
3.3 Development Network With a Central WebSpeed Machine 3–7
3.4 Development Network With a Dedicated Web Server 3–9
3.5 Multiple LAN Development Environment 3–10
3.6 Deployment Configuration With a Dedicated Web Server 3–14
3.7 Development and Deployment In a Shared Configuration 3–16

4. Before You Install WebSpeed 4–1
4.1 Preparing Your Web Server 4–2
4.1.1 Understanding Your Web Server 4–2
4.1.2 Testing Your Network With the ping Command 4–4
4.1.3 Reaching Your Web Server's Home Page 4–4
4.1.4 Creating Directories On Your Web Server 4–4
4.2 Planning Your WebSpeed Installation 4–5
4.2.1 Connection Limits 4–5
4.3 Matching Your Web Server and WebSpeed Messenger 4–6
4.4 Preinstallation Tasks 4–7

5. Installing WebSpeed On Windows 5–1
5.1 Beginning the Installation 5–2
5.2 Performing a Typical Or a Complete Installation 5–8
5.3 Performing a Custom Installation 5–15
5.4 Installing Additional Products 5–29
5.5 Network Installations 5–29

5.6 The InstallShield Silent (Batch Mode) Utility 5–30
5.6.1 Creating the Response File Automatically 5–30
5.6.2 Creating the Response File Manually 5–30
5.6.3 Running the InstallShield Silent Utility 5–33
5.7 Uninstalling WebSpeed 5–34
5.7.1 Before You Begin 5–34
5.7.2 Using the Uninstall Or Add/Remove Program Utility 5–34

6. Installing WebSpeed On UNIX. 6–1
6.1 Beginning the Installation 6–2
6.2 Performing a Typical Or a Complete Installation 6–10
6.2.1 Selecting the Destination Pathname 6–11
6.3 Performing a Custom Installation 6–20
6.4 Running the Installation Utility In Batch Mode 6–31
6.4.1 Installation Initialization File 6–31
6.4.2 Running a Batch Installation 6–33
6.5 Setting AdminServer Security 6–34

7. Configuring WebSpeed On Windows. 7–1
7.1 WebSpeed Configuration Overview 7–3
7.2 WebSpeed Administration 7–4
7.2.1 The AdminServer 7–4
7.2.2 Progress Explorer 7–5
7.2.3 WebSpeed Command-line Utilities 7–6
7.2.4 NameServer Command-line Utilities 7–7
7.3 Setting Up WebSpeed On the Web Server Machine 7–8
7.3.1 Installing the Messenger Executable 7–8
7.3.2 Configuring Netscape Web Servers For NSAPI Messengers 7–10
7.3.3 Restarting the Netscape Web Server 7–12
7.4 Placing Static Files On the Web Server Machine 7–12
7.5 Overview Of the ubroker.properties File 7–13
7.6 Starting the AdminServer 7–13
7.7 Setting Up the WebSpeed Environment 7–14
7.7.1 Setting Environment Variables 7–14
7.8 Configuring WebSpeed Components 7–15
7.8.1 Understanding the NameServer's Load Balancing Option 7–16
7.9 Starting WebSpeed Transaction Server and NameServer Instances 7–18
7.10 Maintaining the WebSpeed Transaction Server and NameServer Log Files 7–20
7.11 Configuring a WebSpeed Messenger-only Installation 7–20
7.12 Starting WebSpeed To Test the Configuration 7–21

7.13 Testing Your Configuration 7–21
7.13.1 Using the Progress Explorer To Check Status 7–21
7.13.2 Using the Query Option To Check Status 7–22
7.13.3 Viewing the Messenger Administration Page 7–23
7.13.4 Running the status.p Procedure 7–23
7.14 Managing the WebSpeed Transaction Server 7–23
7.14.1 Dynamically Starting Additional Agents 7–24
7.14.2 Trimming Running Agents. 7–24
7.14.3 Stopping the Transaction Server. 7–24
7.14.4 Accessing Help On WTBMAN. 7–24
7.15 Managing the WebSpeed Messenger 7–25

8. Configuring WebSpeed On UNIX 8–1
8.1 WebSpeed Configuration Overview 8–3
8.2 WebSpeed Administration 8–4
8.2.1 The AdminServer 8–5
8.2.2 Progress Explorer 8–5
8.2.3 WebSpeed Command-line Utilities 8–5
8.2.4 NameServer Command-line Utilities 8–6
8.3 Setting Up WebSpeed On the Web Server Machine 8–7
8.3.1 Installing the Messenger Executable 8–8
8.3.2 Configuring the Netscape Web Server For the NSAPI Messenger 8–9
8.3.3 Restarting the Netscape Web Server 8–12
8.3.4 Configuring a CGI Messenger 8–12
8.4 Placing Static Files On the Web Server Machine. 8–14
8.5 Overview Of the ubroker.properties File 8–15
8.5.1 The ubroker.properties File Structure 8–15
8.5.2 Editing the ubroker.properties File 8–16
8.6 Starting the AdminServer 8–18
8.7 Setting Up the WebSpeed Environment 8–19
8.7.1 Setting Environment Variables 8–19
8.8 Configuring WebSpeed Components 8–20
8.8.1 Understanding the NameServer's Load Balancing Option 8–21
8.9 Starting WebSpeed Transaction Server and NameServer Instances . . . 8–23
8.10 Maintaining the WebSpeed Transaction Server and NameServer Log Files 8–25
8.11 Configuring a WebSpeed Messenger-only Installation 8–25
8.12 Starting WebSpeed To Test the Configuration 8–26
8.13 Testing Your Configuration. 8–26
8.13.1 Using the Progress Explorer To Check Status 8–27
8.13.2 Using the Query Option To Check Status. 8–27
8.13.3 Viewing the Messenger Administration Page 8–28
8.13.4 Running the status.p Procedure 8–28

8.14 Managing the WebSpeed Transaction Server 8–29
8.14.1 Dynamically Starting Additional Agents 8–29
8.14.2 Trimming Agents 8–29
8.14.3 Stopping the Transaction Server 8–30
8.14.4 Accessing Help On WTBMAN 8–30
8.15 Managing the WebSpeed Messenger 8–30

9. WebSpeed Dynamic Code-page Support: How It Works and How To Configure It 9–1
9.1 Introduction 9–2
9.2 Basic Browser-Agent Interaction 9–2
9.3 How Dynamic Code-page Support Works 9–3
9.4 Summary of Code-page Conversions 9–5
9.5 Configuring the Agent 9–5
9.6 Tasks For the WebSpeed Developer 9–5

10. Connecting To a Data Source 10–1
10.1 Connecting To a Progress Database 10–2
10.1.1 Connecting Programmatically 10–3
10.1.2 Connecting To a Local Database 10–3
10.1.3 Connecting To a Remote Database 10–4
10.2 Connecting To a Non-Progress Data Source Through a DataServer 10–4
10.2.1 DataServer and WebSpeed Architecture 10–6
10.2.2 Setting Up and Running the DataServer 10–8
10.3 Creating an International Web Site 10–9
10.3.1 Setting Up Transaction Servers 10–11
10.3.2 Configuring Agents 10–12
10.3.3 Directing Web Requests 10–14

11. Starting Your WebSpeed Configuration. 11–1
11.1 Startup Options 11–2
11.2 Single-machine Configuration 11–3
11.3 Development Network With a Central WebSpeed Machine 11–4
11.4 Development Network With a Dedicated Web Server 11–5
11.5 Multiple LAN Development Environment 11–6
11.6 Deployment Configuration With a Dedicated Web Server 11–8
11.7 Development and Deployment Shared Configuration 11–9

12. Upgrading WebSpeed On Windows 12–1
12.1 Migrating Your Existing WebSpeed Version 2 Configuration To WebSpeed Version 3.1 12–2
12.2 Coexistence Of WebSpeed Version 2 With WebSpeed Version 3.1 12–6
12.3 Migrating Your Existing WebSpeed Version 3.0 Configuration To WebSpeed Version 3.1 12–9
12.4 Coexistence Of WebSpeed Version 3.0 With WebSpeed Version 3.1 12–10

13. Upgrading WebSpeed On UNIX 13–1
13.1 Migrating Your Existing WebSpeed Version 2 Configuration To WebSpeed Version 3.1 13–2
13.2 Coexistence Of WebSpeed Version 2 With WebSpeed Version 3.1 13–6
13.3 Migrating Your Existing WebSpeed Version 3.0 Configuration To WebSpeed Version 3.1 13–9
13.4 Coexistence Of WebSpeed Version 3.0 With WebSpeed Version 3.1 13–11

A. Progress WebSpeed Directory Structure A–1
A.1 Installation Path Directory Structure A–2

B. Component Lists For WebSpeed Products B–1
B.1 WebSpeed Product Components and Subcomponents B–2
B.1.1 NameServer B–2
B.1.2 NameServer Enterprise Edition B–3
B.1.3 Personal Database B–4
B.1.4 ProVision Plus B–7
B.1.5 WebSpeed Development Server B–13
B.1.6 WebSpeed Enterprise Transaction Server B–16
B.1.7 WebSpeed Messenger B–19
B.1.8 WebSpeed Transaction Server B–19
B.1.9 WebSpeed Workshop B–22

C. Configuration and Management Utilities C–1
C.1 NSCONFIG Utility C–3
C.2 NSMAN Utility C–5
C.3 WSCONFIG Utility C–8
C.4 WTBMAN Utility C–10
C.5 WebSpeed Error Customization Utility C–14
C.6 SHOWCFG Utility C–17
C.6.1 Using the SHOWCFG Utility On Windows C–17
C.6.2 Using the SHOWCFG Utility On UNIX C–19
C.7 Product Update Utility C–20
C.7.1 Using the Product Update Utility On Windows C–21
C.7.2 Using the Product Update Utility On UNIX C–21

Index Index–1

Figures

Figure 2–1:	How WebSpeed Processes an HTML Client Request	2–8
Figure 3–1:	Configuration For Single Developer On a Single Machine	3–5
Figure 3–2:	Development Network With a Central WebSpeed Machine	3–7
Figure 3–3:	Development Network With a Dedicated Web Server	3–9
Figure 3–4:	Multiple LAN Development Environment	3–11
Figure 3–5:	Deployment Configuration With a Dedicated Web Server	3–14
Figure 3–6:	Development and Deployment In a Shared Configuration	3–16
Figure 5–1:	Question Dialog Box	5–25
Figure 5–2:	Warning Dialog Box	5–25
Figure 10–1:	The Local DataServer	10–6
Figure 10–2:	Remote DataServer	10–7
Figure 10–3:	WebSpeed Configuration That Supports an International Web Site	10–10
Figure 11–1:	Configuration For Single Developer On a Single Machine	11–3
Figure 11–2:	Development Network With a Central WebSpeed Machine	11–4
Figure 11–3:	Development Network With a Dedicated Web Server	11–5
Figure 11–4:	Multiple LAN Development Environment	11–6
Figure 11–5:	Deployment Configuration With a Dedicated Web Server	11–8
Figure 11–6:	Development and Deployment Shared Configuration	11–9
Figure C–1:	Utility Syntax Conventions	C–2
Figure C–2:	WebSpeed Error Message Customization Utility	C–15
Figure C–3:	Product Configuration Information From the Windows Version Of the SHOWCFG Utility	C–18
Figure C–4:	Product Configuration Displayed Using the SHOWCFG Utility On UNIX	C–20

Tables

Table 1–1: Hardware and Networking Requirements 1–3
Table 1–2: JRE/JDK Installations 1–4
Table 1–3: WebSpeed Component Requirements For Windows 1–6
Table 1–4: Approximate Disk Space On Windows 1–8
Table 1–5: Approximate Disk Space On UNIX 1–9
Table 1–6: WebSpeed Agent/Database Server Compatibility 1–10
Table 1–7: Displaying Product License Information 1–11
Table 2–1: WebSpeed Command-line Utilities 2–6
Table 4–1: About Your Web Server 4–2
Table 4–2: Database Connection Limits 4–5
Table 4–3: WebSpeed Agent Limits For Transaction Servers 4–6
Table 4–4: Web Server and Messenger Types 4–6
Table 4–5: Preinstallation Tasks 4–7
Table 5–1: Illegal Characters 5–24
Table 6–1: Mount Commands 6–3
Table 6–2: Product Configuration Data Dialog Box Options 6–5
Table 6–3: Type, Device and Destination Dialog Box Options 6–9
Table 6–4: Installation Options 6–10
Table 6–5: Destination Pathname Exists Dialog Box Options 6–12
Table 6–6: Custom – Select Products Dialog Box Status Information 6–21
Table 6–7: Destination Pathname Exists Dialog Box Options 6–22
Table 6–8: User-Group Parameter Options 6–34
Table 7–1: Windows NT WebSpeed Messengers 7–9
Table 7–2: Netscape Web Server Configuration 7–10
Table 7–3: Weight Factors Based On Percentage 7–16
Table 7–4: Weight Factors Based On Arbitrary Sums 7–17
Table 8–1: UNIX WebSpeed Messengers 8–8
Table 8–2: Netscape Web Server Configuration 8–9
Table 8–3: NSAPI Messenger Settings 8–11
Table 8–4: Options For Invoking the CGI Messenger 8–13
Table 8–5: Weight Factors Based On Percentage 8–21
Table 8–6: Weight Factors Based On Arbitrary Sums 8–22
Table 10–1: Remote Connection Parameters 10–4
Table 12–1: WebSpeed Version 2 To Version 3.1 Conversion Chart For Windows NT 12–3
Table 13–1: WebSpeed Version 2 To Version 3.1 Conversion Chart For UNIX 13–3
Table A–1: Progress (%DLC%) Directory Structure A–2
Table B–1: NameServer Components and Subcomponents B–2
Table B–2: NameServer Enterprise Edition Components and Subcomponents B–3
Table B–3: Personal Database Components and Subcomponents B–4
Table B–4: ProVision Plus Components and Subcomponents B–7

Table B–5: WebSpeed Development Server Components and Subcomponents . . B–13
Table B–6: WebSpeed Enterprise Transaction Server Components and Subcomponents B–16
Table B–7: WebSpeed Messenger Components B–19
Table B–8: WebSpeed Transaction Server Components and Subcomponents B–19
Table B–9: WebSpeed Workshop Components and Subcomponents B–22
Table C–1: Progress Command Components C–2
Table C–2: NSCONFIG Command Examples C–4
Table C–3: NSMAN Command Examples C–6
Table C–4: WSCONFIG Command Examples C–9
Table C–5: WTBMAN Command Examples C–12

Preface

Purpose

The *WebSpeed Installation and Configuration Guide* discusses planning, installing, and configuring WebSpeed on both Windows and UNIX systems.

Audience

This guide is for anyone who is creating a WebSpeed environment on either a Windows or a UNIX system. You must have administrative privileges on the system on which you are installing WebSpeed components. You should have a working knowledge of Web servers and networking issues before installing WebSpeed.

Organization Of This Manual

Chapter 1, "Installation Requirements"

Provides information on system requirements, product licensing, and product life cycles.

Chapter 2, "Overview Of the WebSpeed Architecture"

Describes the components of the WebSpeed environment and how they work together. It also introduces the Progress administration framework.

Chapter 3, "WebSpeed Configurations"

Discusses the options available to you for designing distributed installations of WebSpeed. The chapter includes several configuration examples that illustrate how WebSpeed can be distributed across your network.

Chapter 4, “Before You Install WebSpeed”

Describes the tasks you perform before installing WebSpeed.

Chapter 5, “Installing WebSpeed On Windows”

Provides step-by-step instructions for installing WebSpeed on Windows systems.

Chapter 6, “Installing WebSpeed On UNIX”

Provides step-by-step instructions for installing WebSpeed on UNIX systems.

Chapter 7, “Configuring WebSpeed On Windows”

Provides information on how to configure a WebSpeed environment on a Windows system. It also includes information about the Progress Explorer, command-line utilities, and the WebSpeed property file, `ubroker.properties`.

Chapter 8, “Configuring WebSpeed On UNIX”

Provides information on how to configure a WebSpeed environment on a UNIX system. It also includes information about the Progress Explorer, command-line utilities, and the WebSpeed property file, `ubroker.properties`.

Chapter 9, “WebSpeed Dynamic Code-page Support: How It Works and How To Configure It”

Explains how WebSpeed dynamic code-page support works and how to configure it.

Chapter 10, “Connecting To a Data Source”

Describes how to connect WebSpeed Agents to a Progress or non-Progress data source for application development. The chapter also describes how to set up an international WebSpeed configuration.

Chapter 11, “Starting Your WebSpeed Configuration”

Discusses how to properly start up distributed WebSpeed environments. The chapter uses the same configuration examples used in Chapter 3, “WebSpeed Configurations,” to illustrate the correct startup sequence of the WebSpeed components.

Chapter 12, “Upgrading WebSpeed On Windows”

Describes how to convert earlier versions of WebSpeed to WebSpeed Version 3.1 on Windows systems. The chapter also discusses whether earlier versions can coexist with WebSpeed Version 3.1. For information about converting databases from Progress Version 8 to Version 9, see the *WebSpeed Product Update Bulletin*.

Chapter 13, “Upgrading WebSpeed On UNIX”

Describes how to convert earlier versions of WebSpeed to WebSpeed Version 3.1 on UNIX systems. The chapter also discusses whether earlier versions can coexist with WebSpeed Version 3.1. For information about converting databases from Progress Version 8 to Version 9, see the *WebSpeed Product Update Bulletin*.

Appendix A, “Progress WebSpeed Directory Structure”

Lists the subdirectories of your installation path.

Appendix B, “Component Lists For WebSpeed Products”

Provides a list of the installation components and subcomponents for each WebSpeed product.

Appendix C, “Configuration and Management Utilities”

Provides information on the WSCONFIG, WTBMAN, NSCONFIG, and NSMAN, SHOWCFG, and Product Update utilities. The entries provide the full syntax, explanation of options, examples, and notes on using each utility.

Windows and UNIX Path Conventions

Because many users create WebSpeed environments that have both Windows and UNIX machines, the *WebSpeed Installation and Configuration Guide* covers both operating systems. Whenever the only difference between a command on Windows and UNIX is the use of back slashes or forward slashes in the path, this manual uses the Windows path convention.

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 ... ] }
       }
```

WebSpeed Messages

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

- Execution messages inform you of errors encountered while WebSpeed is running a procedure (for example, if it cannot find a record with a specified index field value).
- Compile messages inform you of errors found while WebSpeed 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 WebSpeed is getting ready to execute (for example, if you entered an invalid startup parameter).

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

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

Use the WebTools Message tool to see more information about WebSpeed messages. In addition, the online help for many Progress tools includes the following Help menu options:

- Choose Help→ Recent Messages to display detailed descriptions of the most recent WebSpeed message and all other messages returned in the current session.
- Choose Help→ Messages, then enter the message number to display a description of any WebSpeed message. (If you encounter an error that terminates WebSpeed, make a note of the message number before restarting.)

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 Installation and Configuration Guide Version 9 for Windows

A manual that describes how to install and set up Progress Version 9.1 for all supported Windows and Citrix MetaFrame operating systems.

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 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 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.

1

Installation Requirements

WebSpeed allows you to develop Web-based Internet Transaction Processing (ITP) applications: business applications that support real-time data access and data management.

This guide tells you how to install and configure WebSpeed on machines running supported Windows or UNIX operating systems. This chapter covers the following topics:

- Quick Tour Of the WebSpeed Components
- System Requirements
- WebSpeed Agent/Database Server Compatibility
- WebSpeed Licensing Information
- Product Life Cycle

1.1 Quick Tour Of the WebSpeed Components

In order to correctly calculate the required system resources, you need to be familiar with the WebSpeed components. The following list gives a brief description of the function of each component:

- **WebSpeed Workshop**—The WebSpeed WorkShop allows you to develop and test WebSpeed applications. It includes the WebSpeed AppBuilder, PRO*Tools, the WebSpeed WebTools, and the WebSpeed Development Server.
- **WebSpeed Transaction Server**—The WebSpeed Transaction Server accepts Web requests from an HTML client, processes them, and returns the results as HTML. The WebSpeed Transaction Server is made up of the following components:
  - **WebSpeed Agent**—The WebSpeed Agent accepts Web requests from a WebSpeed Messenger, processes them, and returns results as HTML.
  - **WebSpeed Broker—**The WebSpeed Broker maintains a pool of WebSpeed Agents and manages connection requests from a WebSpeed Messenger.
  - **WebSpeed Messenger**—The WebSpeed Messenger handles the transfer of data between the Web server and the WebSpeed Agent during a single Web transaction.
- **NameServer**—The NameServer maintains a list of WebSpeed Transaction Servers, directing a WebSpeed Messenger to one that can handle its request.
- **Progress Administration Framework—**All Progress server products use a common administration framework. This framework consists of the AdminServer, the Progress Explorer, and the Management Utilities.

For more detailed descriptions of the components, see the "WebSpeed Components" section in Chapter 2, "Overview Of the WebSpeed Architecture."

1.2 System Requirements

You build a WebSpeed configuration around your system's hardware, its networking, and the WebSpeed components. WebSpeed configurations can be on a single machine or distributed across a network. The following sections cover the hardware, network, and software that your system requires before you can install WebSpeed.

1.2.1 Hardware and Networking Requirements

Table 1–1 lists the hardware and networking requirements for creating a WebSpeed configuration on your system. Depending on how you create your applications, you might have different requirements when you deploy your applications.

Table 1–1: Hardware and Networking Requirements

Component	Requirement
CD-ROM drive	You need CD-ROM drives that are compatible with each operating system on which you will install WebSpeed components.
VGA or higher resolution video adapter	Your video adapter should be running at 800x600 resolution. For the best visual effect, your system should support 256 colors. This is only necessary for machines on which you install the WebSpeed Workshop, ProVision Plus, or the Progress Explorer.
Network access	Your system must have access to an Intranet or the Internet.
Network protocol	Your system must support the TCP/IP and UDP protocols. The platforms currently supported by WebSpeed provide built-in TCP/IP and UDP support.
Web server	Your Web server must support one of the following interfaces: • **ISAPI—**For example, Microsoft Internet Information Server (IIS), Versions 3.x, 4.x, or 5.x • **NSAPI**—For example, iPlanet Enterprise or Fast Track Server, Version 3.x or 4.0x • **CGI 1.1**—For example, Microsoft Internet Information Server (IIS), Versions 3.x, 4.x, or 5.x; Apache; iPlanet Enterprise or Fast Track Server, Version 3.x, 4.x, or 6.x.
Web browser	To use the new browser-based editor, you need one of the following Web browsers to run the WebSpeed Workshop: • Microsoft Internet Explorer, Version 5.0 or later. The version of Internet Explorer you are using must support JScript Version 2.0. • Netscape Navigator, Version 4.5 or later.

1.2.2 Java Requirements

WebSpeed Version 3.1 requires one or more installed Java components for internal use. Specifically, the components are the Java Runtime Environment (JRE) and the Java Development Kit (JDK). On most operating systems, Progress installs the required JRE and JDK automatically with the development products. Table 1–2 lists the JRE and JDK requirements for the platforms that WebSpeed supports.

Table 1–2: JRE/JDK Installations *(1 of 2)*

Platform	JRE	JDK	JDK Download Location
Windows NT	Installed by Progress	Installed by Progress[1]	http://www.microsoft.com
Windows 95	Installed by Progress	Installed by Progress[1]	http://www.microsoft.com
Windows 98	Installed by Progress	Installed by Progress[1]	http://www.microsoft.com
Windows 2000	Installed by Progress	Installed by Progress[1]	http://www.microsoft.com
Windows XP Professional	Installed by Progress	Installed by Progress[1]	http://www.microsoft.com
Sun Solaris SPARC 2.6, 2.7 (32-bit)	Installed by Progress	Installed by Progress[1]	http://www.sun.com
Sun Solaris SPARC 8 (64-bit)	Installed by Progress	Installed by Progress[1]	http://www.sun.com
HP-UX 11 (32-bit)	Installed by Progress	HP Version 1.3.0 (optional)	http://www.hp.com
HP-UX 11 (64-bit)	Installed by Progress	HP Version 1.3.0 (optional)	http://www.hp.com
IBM AIX 4.3.2, 4.3.3 (32-bit)	Version 1.3.0	Version 1.3.0	http://www.ibm.com

Table 1–2: JRE/JDK Installations *(2 of 2)*

Platform	JRE	JDK	JDK Download Location
SCO UnixWare 7	SCO Version 1.3.0_02	SCO Version 1.3.0_02	http://www.sco.com
Compaq Tru64 UNIX	Compaq Version 1.3.1	Compaq Version 1.3.1	http://www.compaq.com
Linux 6.2, 7.0 (Red Hat)	Version 1.3.0_02	Version 1.3.0_02	www.java.sun.com/products/
Linux 7.1 and later (Red Hat Linux, TurboLinix, Caldera OpenLinux, LinuxIT SuSE Linux)	Version 1.3.1	Version 1.3.1	www.java.sun.com/products/

[1] Progress installs the JDK with its development products: ProVision Plus, ProVision Plus Development Server, WebSpeed Workshop, and WebSpeed Development Server.

NOTES: There is an exception to the JRE/JDK installation information in Table 1–2 for Windows NT. If you are working in a Japanese environment on Windows NT and want to receive Japanese messages for Java developer tools like the Java compiler, Javadoc, and AppletViewer, you should first install the JavaSoft Japanese Java Software Development Kit (JDK) supplement.

The default version of the JRE installed by Progress is 1.3.0.

The default version of the JDK installed by Progress is 1.3.0.

If you install the WebSpeed Messenger on a different machine from the WebSpeed Transaction Server, there are no Java requirements on the machine where the Messenger resides.

1.2.3 WebSpeed Component Requirements On Windows

The WebSpeed Transaction Server and the WebSpeed Workshop is supported on Windows NT 4.0, 2000, and XP on Intel processors. On Windows 95, 98, or ME, you can only develop applications with WebSpeed Workshop.

Different WebSpeed components place different demands on your system resources. A workstation running the WebSpeed AppBuilder can operate efficiently with a slower processor than a machine running the WebSpeed Transaction Server can. The startup parameters that you set for a WebSpeed component can change the amount of RAM a machine needs.

Table 1–3 lists the minimum requirements for running a WebSpeed configuration on a Windows NT 4.0 network. The amount of RAM you need varies by configuration. This table lists only the minimum amount of RAM that you need for the operating system. Your configuration might need more RAM. See the "Disk Space Requirements" section for information about the amount of disk space needed for your installation.

When reading this table, look at the most resource-intensive component you plan to install on a particular machine. For example, a Complete Installation of the WebSpeed Workshop product installs the WebSpeed Development Server as well as the WebSpeed AppBuilder. Because of that, you need to install it on a machine that meets the requirements for the WebSpeed Development Server, rather than the WebSpeed AppBuilder.

Table 1–3: WebSpeed Component Requirements For Windows *(1 of 2)*

Component	Windows System Requirements
WebSpeed AppBuilder	• Windows 95, 98, NT 4.0, XP, ME, 2000 operating system. • A Pentium PC (or compatible computer) with a clock speed of at least 100 MHZ. • 128 MB of RAM.
WebSpeed Messenger	• Windows NT 4.0 (Workstation or Server) with Service Pack 6a[1] or later, Windows XP, Windows 2000. • A Pentium PC (or compatible computer) with a clock speed of at least 100 MHZ. • 128MB of RAM.

Table 1–3: WebSpeed Component Requirements For Windows *(2 of 2)*

Component	Windows System Requirements
WebSpeed Transaction Server	• Windows NT 4.0 (Workstation or Server) with Service Pack 6a[1] or later, Windows XP, Windows 2000. • A Pentium PC (or compatible computer) with a recommended clock speed of at least 200 MHZ. • 128MB of RAM (256MB recommended). Memory requirements vary depending on the number of WebSpeed Agents you are running.
ProVision Plus Development Server	• Windows NT 4.0 (Workstation or Server) with Service Pack 6a[1] or later, Windows XP, Windows 2000. • A Pentium PC (or compatible computer) with a recommended clock speed of at least 200 MHZ. • 128 MB of RAM (256 MB recommended). Memory requirements vary depending on the number of WebSpeed Agents you are running.
NameServer	• Windows NT 4.0 (Workstation or Server) with Service Pack 6a[1] or later, Windows XP, Windows 2000. • A Pentium PC (or compatible computer) with a recommended clock speed of at least 200 MHZ. • 128 MB of RAM (256MB recommended). Memory requirements vary depending on the number of WebSpeed Agents you are running.

[1] You can download the required service pack from `www.microsoft.com`.

Table 1–3 lists the necessary components of a WebSpeed configuration. If you are adding other Progress products to your system, such as the Progress Database or a DataServer, see the "Installation Requirements" chapter in the *Progress Installation and Configuration Guide Version 9 for Windows*.

WebSpeed can access a variety of databases through the Progress DataServer architecture. There are native DataServers for accessing the ORACLE RDBMS and DB2/400 data on AS/400 systems. The Progress DataServer for ODBC allows applications to access DB2 data on MVS, NT, RS6000, Informix OnLine, Microsoft Access, Microsoft SQL Server, and SYBASE.

Disk Space Requirements

Table 1–4 lists the Progress products commonly used in WebSpeed configurations and the approximate disk space that each 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 block size of your machine. If you are adding other Progress products to your system, such as the Progress Database or a DataServer, see the "Installation Requirements" chapter in the *Progress Installation and Configuration Guide Version 9 for Windows*.

Additionally, some products share common files. If you install several products on your machine, the installation might not require the sum of the values listed in Table 1–4 for each product.

NOTE: During installation, WebSpeed adds files to the Windows directory. Make sure that you have sufficient disk space on the drive where Windows NT is installed.

Table 1–4: Approximate Disk Space On Windows

Product	Total File Size
Name Server	43MB
Name Server (Enterprise)	43MB
Personal Database	205MB
ProVision Plus	518MB
ProVision Plus Development Server	195MB
WebSpeed Development Server	263MB
WebSpeed Messenger	12MB
WebSpeed Transaction Server	198MB
WebSpeed Transaction Server (Enterprise)	198MB
WebSpeed Workshop	422MB

1.2.4 WebSpeed Component Requirements On UNIX

This section discusses disk space and other system requirements for running WebSpeed on your UNIX system. WebSpeed requires no special swap space requirement or shared memory and kernel configurations. If you are using the Progress database, check the *Progress Installation and Configuration Guide Version 9 for UNIX* for information on setting these items.

File Descriptors

You must allocate sufficient file descriptors to handle all the WebSpeed Agents your configuration uses. You should set the file descriptors to 241 by entering the following command:

```
ulimit -n 240
```

Disk Space Requirements

Table 1–5 lists the approximate file size that each Progress product requires based on a 512-byte block size and a complete installation that includes support for one installed language. Actual space requirements can vary, depending on the block size of your machine.

If you are adding other Progress products to your system, such as the Progress Database or a DataServer, see the "Installation Requirements" chapter in the *Progress Installation and Configuration Guide Version 9 for UNIX*.

Additionally, some products share common files. If you install several products on your machine, the installation might not require the sum of the values listed in Table 1–5 for each product.

NOTE: During the installation process, your system requires extra space to contain the installation utility, installation tables, and Progress files in compressed format. This additional space is needed only for the installation. When the installation ends, these extra files are deleted from the current working directory.

Table 1–5: Approximate Disk Space On UNIX *(1 of 2)*

Product	Total File Size
NameServer	24MB
NameServer (Enterprise)	24MB
Personal Database	171MB
ProVision Plus Development Server	212MB

Table 1–5: Approximate Disk Space On UNIX *(2 of 2)*

Product	Total File Size
WebSpeed Development Server	212MB
WebSpeed Messenger	10MB
WebSpeed Transaction Server	176MB
WebSpeed Transaction Server (Enterprise)	176MB

1.3 WebSpeed Agent/Database Server Compatibility

Table 1–6 lists the WebSpeed Agent versions and the versions of the Progress Database Server with which they are compatible.

Table 1–6: WebSpeed Agent/Database Server Compatibility

WebSpeed Agent	Database Server
WebSpeed Versions 2.0 and 2.1	Progress Version 8.x
WebSpeed Version 3.0	Progress Version 8.x and Version 9.x
WebSpeed Version 3.1	Progress Version 8.x and Version 9.x

1.4 WebSpeed Licensing Information

When you install WebSpeed, the installation utility prompts you to enter product information from your *License Addendum*. The installation utility records the license information in the configuration file, `progress.cfg`. Table 1–7 lists the ways you can display the product license information after you have installed WebSpeed.

Table 1–7: Displaying Product License Information

Starting from . . .	Do this . . .
Windows desktop	Choose Start→ Programs→ Progress→ Config.
Command line	Enter the following command: *install-path*/bin/showcfg *install-path*/progress.cfg

Your license allows you to run the WebSpeed Transaction Server on one machine only. There is no restriction on the number of WebSpeed Brokers that you can run concurrently. Note, however, that the maximum number of WebSpeed Agents you are licensed to run refers to the total number of agents being managed by all brokers at any time.

For more information about licensing and setting up WebSpeed Brokers and WebSpeed Agents, see Chapter 7, “Configuring WebSpeed On Windows,” or Chapter 8, “Configuring WebSpeed On UNIX.”

1.5 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. The Product Life Cycle 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 the 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 document does not address Progress Software's future product plans. Please be advised that the information contained in this document is subject to change by Progress Software without notice. For further information, please contact your Progress Software representative or visit The Progress Software Web site at `www.progress.com`.

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 provide BasicService by working with customers towards an appropriate resolution. Progress Software will evaluate certifying Active Phase products with new hardware platforms, operating systems, and third-party databases.

- **Functionally Stable**—Functionally Stable Phase products are fully supported but do not receive new features. If product problems are found, Progress Software Technical Support will provide BasicService by working with customers towards an appropriate resolution. Progress Software will evaluate certifying Functionally Stable Phase products with new hardware platforms, operating systems, and third-party databases.

- **Mature**—Mature Phase products do not receive new features and are supported in a limited fashion. 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 certified with new hardware platforms, operating systems, or third-party databases.

  Due to the advanced age of Mature Phase products, Progress Software provides only Limited BasicService. The skills and resources required for supporting Mature Phase products diminish over time. 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 licenses that do not require the delivery of media or documentation. You can continue to develop and deploy applications with licensed Retired Phase products. Progress Software Corporation will NOT certify Retired Phase products with new hardware platforms, operating systems, or third-party databases. Progress Software provides 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.

  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

Overview Of the WebSpeed Architecture

In order to install and configure WebSpeed, you must understand the overall WebSpeed architecture and how the various components of the architecture work together. Most of these components are unique to WebSpeed; however, the AdminServer and NameServer serve as a common administration mechanism for WebSpeed as well as for the Progress AppServer, Database, and DataServer products.

This chapter provides information about the following topics related to WebSpeed architecture:

- WebSpeed Components
- The Progress Administration Framework
- The ubroker.properties File
- How WebSpeed Components Work Together

2.1 WebSpeed Components

A WebSpeed installation consists of several components that you can install on a single machine or that you can distribute across several machines. The WebSpeed components are:

- **WebSpeed Agent**—An application process that executes the Web objects, performs database transactions, and dynamically merges data into HTML format. This is the standard character Progress 4GL client running in batch mode.
- **WebSpeed Broker**—An application that does the following:
  - Registers the WebSpeed Services that it provides to a NameServer for access by one or more HTML clients. The HTML client runs from within an Internet browser.

    If you are interested in running WebSpeed from a client other than the HTML client (for example, an ActiveX page or a Java application), see the "Interfaces" area on the Developer's Corner Web site at `http://developer.webspeed.com`.
  - Manages connections between clients and a pool of WebSpeed Agents.
  - Maintains the status of each agent in its pool and dynamically scales the number of agents according to changing demand.

  NOTE: If you start a WebSpeed Broker without specifying a username, the Broker inherits the account that the AdminServer is using. This is generally the system account, which might not have access to network drives.
- **WebSpeed Messenger**—A process that handles the transfer of data between the Web server and the WebSpeed Agent during a single Web transaction. The Messenger is a CGI program or an ISAPI or NSAPI process, depending on the Web server and how you plan to write your applications.

  There is also a Messenger that works with Microsoft's Active Server Pages, the WSASP Messenger. Using the WSASP Messenger, you can call out of an Active Server Page to a WebSpeed application.

- **NameServer—**The NameServer maintains a list of WebSpeed Transaction Servers. The Transaction Servers register the application services they provide with the NameServer. The NameServer can then direct client connection requests to a WebSpeed broker that supports a requested application service. This provides scalability and location transparency to your applications.

  When the Enterprise Transaction Server is installed, the NameServer also performs load balancing. Load balancing allows you to balance client workload among multiple brokers that support the same application service (that is, the same set of procedures and resources).

- **WebSpeed Workshop Environment**—The WebSpeed WorkShop Environment is made up of the tools that you use to develop and test WebSpeed applications. It includes the following:

  – **WebSpeed AppBuilder**—The WebSpeed AppBuilder is the visual programming environment in which you develop applications for the Web. The AppBuilder unifies traditional client/server and n-tier functionality with Web-based technology to support a broad, integrated range of application and development options.

    If you purchased ProVision Plus, the combined Progress and WebSpeed product, you have the ProVision Plus AppBuilder. The ProVision Plus AppBuilder allows you to build graphical user interfaces, as well as browser-based user interfaces, for your Progress applications.

    Either version of the AppBuilder runs on Windows 95, Windows 98, and Windows NT 4.0. You can install it on a Windows machine and configure it to work with a WebSpeed Development Server on a UNIX machine.

    (For an introduction to using the AppBuilder, see *Getting Started with WebSpeed* or the *Progress Application Development Environment — Getting Started* guide.)

  – **PRO*Tools**—PRO*Tools is a set of utility programs related to the process of developing and running Progress applications. For example, one of the PRO*Tools allows you to edit your PROPATH, the path Progress searches to find programs. When you select PRO*Tools from the Tools menu, the ADE displays a resizable palette of icons. You then run an individual PRO*Tool by clicking on its icon. The Color Changer, Screen Scaling Utility, and ProtoGen PRO*Tools do not apply to WebSpeed.

  - **WebTools**—You use the browser-based WebTools to access information on your server, such as the status of CGI Variables. You can also access database information, use the WebSpeed File tools, and access ProMon reports. You can use the Scripting Lab to write and test WebSpeed code, such as HTML that includes Embedded SpeedScript, and send operating system commands. With the Editor WebTool, you can create, open, save, and print files; check syntax; and compile code.

  - **WebSpeed Development Server**—This version of the WebSpeed Transaction Server is intended to support a single developer's efforts.

- **WebSpeed Transaction Server Products**—There are three WebSpeed Transaction Server products:

  - **WebSpeed Transaction Server—**This version of the WebSpeed Transaction Server includes the AdminServer, the NameServer, WebSpeed Messengers, WebSpeed Brokers, and five or twenty-five WebSpeed Agents. WebSpeed Brokers launch WebSpeed Agents and drive your Web applications. The greater number of available agents allows you to scale to larger applications. A NameServer working with the WebSpeed Transaction Server provides location transparency.

  - **WebSpeed Enterprise Transaction Server**—This version of the WebSpeed Transaction Server includes the AdminServer, the NameServer, WebSpeed Messengers, WebSpeed Brokers, and supports fifty or two hundred fifty WebSpeed Agents. The greater number of available agents allows you to scale to larger applications. An Enterprise NameServer working with the WebSpeed Enterprise Transaction Server provides location transparency and load balancing.

  - **WebSpeed Development Server**—This version of the WebSpeed Transaction Server includes the AdminServer, the NameServer, WebSpeed Messengers, and WebSpeed Brokers. However, it has only two WebSpeed Agents and is intended to support a single developer's efforts. The Development Server allows you, through a browser, to run and test applications you build in the WebSpeed WorkShop so that you can evaluate how they perform when served by your Web server and processed by WebSpeed Agents.

Which version of the WebSpeed Transaction Server you should install depends on your needs. The WebSpeed Development Server provides two agents and supports the development efforts of one user only. The WebSpeed Transaction Server and the WebSpeed Enterprise Transaction Server are both effective for team development or deployment. The WebSpeed Enterprise Transaction Server also supports heavier loads than the WebSpeed Transaction Server.

The WebSpeed installation provides one predefined WebSpeed Broker and one predefined NameServer. You can use these predefined components as templates from which you create and configure additional instances of the WebSpeed Broker and, if needed, the NameServer.

2.2 The Progress Administration Framework

Progress server products use a common administration framework. This framework consists of the AdminServer, the Progress Explorer, and the Management Utilities.

2.2.1 AdminServer

An AdminServer is installed on every system where you install a Progress database, ODBC or ORACLE DataServer, AppServer, NameServer, or WebSpeed Transaction Server. The AdminServer must be running in order to use Progress Explorer or the command-line configuration utilities. These tools contact the AdminServer to perform the actual configuration of your Progress server products.

On Windows NT-based systems, the AdminServer starts automatically and runs as an NT service. For UNIX-based systems, a command-line utility (PROADSV) is used to start and stop the AdminServer.

NOTE: For the complete syntax of the PROADSV utility see the “Setting Up Your Environment” chapter of the *Progress Installation and Configuration Guide Version 9 for UNIX*.

2.2.2 Progress Explorer

The Progress Explorer is a graphical user interface tool that runs as a plugin to the Microsoft Management Console (MMC) on Windows. You can start the Progress Explorer from the Start menu on the Windows taskbar. With the Progress Explorer, you can do the following:

- Create, start, stop, and query the status of NameServers and WebSpeed Transaction Servers.
- Configure NameServers, WebSpeed Transaction Servers, and WebSpeed Messengers.
- Start additional or trim back running WebSpeed Agents.

You can use the Progress Explorer from any Windows machine to administrate remote WebSpeed installations on either Windows or UNIX machines. For more information about using the Progress Explorer to create or configure WebSpeed components, see Chapter 7, "Configuring WebSpeed On Windows," and the Progress Explorer online help.

2.2.3 Command-line Utilities

The command-line utilities allow you to manage existing WebSpeed component configurations, as described in Table 2–1.

Table 2–1: WebSpeed Command-line Utilities

Utility	Function
WTBMAN	Start a WebSpeed Transaction Server, query its status, start and stop additional WebSpeed Agents, trim by a certain number of agents, and stop the WebSpeed Transaction Server.
WSCONFIG	Validate WebSpeed Transaction Server and WebSpeed Messenger properties files.
NSMAN	Start a NameServer, query its status, and stop the NameServer.
NSCONFIG	Validate NameServer properties files.

2.3 The ubroker.properties File

All of the administrative functions that you can perform by using the Progress Explorer or the command-line utilities rely on the AdminServer to interact with other WebSpeed components. This interaction is based on configuration information maintained in the WebSpeed properties file (`ubroker.properties`) that is installed with WebSpeed. The `ubroker.properties` file is the property file for the WebSpeed Broker, WebSpeed Messengers, and the NameServer. The `ubroker.properties` might also contain configuration information for AppServers and DataServers.

The `ubroker.properties` file resides in the *`install-path`*`\properties` directory and includes comments relevant to setting properties for your WebSpeed configuration.

For information about changing the WebSpeed configuration by editing the `ubroker.properties` file, see the comments included in the property file. For an overview of this file, see the "Configuring Progress Unified Broker Products" chapter in the *Progress Installation and Configuration Guide Version 9 for Windows* and the "Overview Of the ubroker.properties File" section in Chapter 7, "Configuring WebSpeed On Windows," or Chapter 8, "Configuring WebSpeed On UNIX."

2.4 How WebSpeed Components Work Together

In the configuration shown in Figure 2–1, the Web server and the WebSpeed Messenger are on the same machine, as required. The AdminServer, the NameServer, the WebSpeed Broker, and the WebSpeed Agents reside on a second machine.

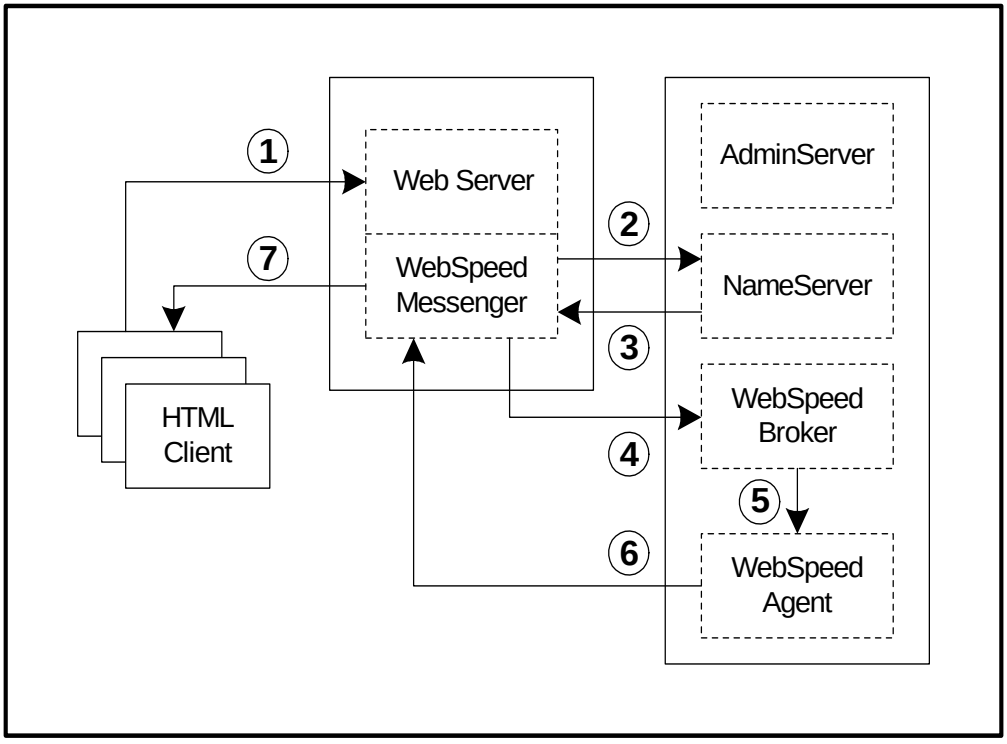


Figure 2–1: How WebSpeed Processes an HTML Client Request

In Figure 2–1, you can follow the path of a WebSpeed transaction, beginning with a connection request from an HTML Client and continuing as follows:

1. The HTML Client, running in an Internet browser, generates connection requests. Each request is in the form of a URL and is sent to a Web server, which forwards it to a WebSpeed Messenger.

2. The WebSpeed Messenger, which is either a CGI program or an ISAPI or NSAPI process (depending on the Web server and the configuration), sends a request to the NameServer for an available WebSpeed Transaction Server that supports the required application service.

3. The NameServer selects a WebSpeed Transaction Server, which supports the requested application service, from its pool of Transaction Servers and notifies the WebSpeed Messenger which Transaction Server to use.

4. The WebSpeed Messenger forwards the HTML Client's request to the WebSpeed Transaction Server identified by the NameServer.

5. In the WebSpeed Transaction Server, the WebSpeed Broker selects an available WebSpeed Agent from the pool of agents.

6. The WebSpeed Agent executes the HTML Client's request and creates an HTML page that it returns to the WebSpeed Messenger.

7. The WebSpeed Messenger passes the HTML page to the Web server, which passes it back to the HTML Client.

All of these components (the Web server, the WebSpeed Messenger, the WebSpeed Broker, the WebSpeed Agents, and the NameServer) can reside on a single physical machine. However, you can also distribute them on separate machines, with the following restrictions:

- The Web server and the WebSpeed Messenger must reside on the same physical machine.
- The NameServer can reside on any machine, but requires an AdminServer.
- The Transaction Server (the WebSpeed Broker and the WebSpeed Agents) requires an AdminServer on its machine. The WebSpeed Broker and the WebSpeed Agents that it supports must reside on the same physical machine. You can have WebSpeed Brokers and Agents on several machines if you purchase multiple Transaction Servers. For example, if you install the Transaction Server on two machines, you can configure a pool of WebSpeed Agents on each machine. Each pool is managed by its own WebSpeed Broker.

See Chapter 3, "WebSpeed Configurations," for several ways to distribute the components across a network. You can also distribute the databases over a network. For more information about distributing Progress databases across a network, see the *Progress Database Administration Guide and Reference*. For information on accessing a non-Progress data source, see the appropriate Progress DataServer guide. When you read these manuals, which describe a typical client/server environment, substitute the term Agent for Client.

3

WebSpeed Configurations

You can distribute the components required to run WebSpeed in many ways. This chapter describes several sample configurations for WebSpeed that you can expand or modify to meet your own needs. Specifically, this chapter discusses:

- Configuration Choices
- Single-machine Configuration
- Development Network With a Central WebSpeed Machine
- Development Network With a Dedicated Web Server
- Multiple LAN Development Environment
- Deployment Configuration With a Dedicated Web Server
- Development and Deployment In a Shared Configuration

3.1 Configuration Choices

To choose the proper configuration to install, you need to consider your network resources and whether you want to create a development, deployment, or combined development and deployment configuration.

In Chapter 2, "Overview Of the WebSpeed Architecture," the "How WebSpeed Components Work Together" section details how information is passed between various WebSpeed components during a round trip from the HTML client to the WebSpeed Agent and back. The common administration mechanism provided by the AdminServer lets you spread the WebSpeed components across your network. This flexibility gives you choices in how to set up your WebSpeed configuration.

The sample configurations in this chapter illustrate which components must reside on the same machine and how to establish links between the WebSpeed components on different machines. You can adapt these examples to your needs.

3.1.1 Installing the Progress Explorer

The sample configurations in this chapter do not show the Progress Explorer. You can install the Progress Explorer on any Windows machine as a plug-in to the Microsoft Management Console (MMC). You can use the Progress Explorer to configure, start, and stop Progress databases, NameServers, and WebSpeed Transaction Servers on any local or remote machine in your network (NT or UNIX).

You can install the Progress Explorer on any Windows machine, even if you are not installing any other Progress products on the machine. Perform a Custom Installation on that machine and select only the Progress Explorer Tools component.

3.1.2 WebSpeed WorkShop Or ProVision Plus

Your configuration can use the WebSpeed Workshop or ProVision Plus. The WebSpeed Workshop is meant for developers who want to design Web-based applications. The WebSpeed AppBuilder in the WebSpeed Workshop gives you the tools to create applications using SpeedScript that use a Web browser for their user interface. You cannot use the WebSpeed AppBuilder to create graphical user interfaces for your applications.

ProVision Plus is meant for developers who want to design Progress applications with non-Web-based interfaces, in addition to Web-based applications. This product gives you an AppBuilder that, unlike the WebSpeed AppBuilder, can use the complete Progress 4GL. This lets you build graphical interfaces and use the full range of SmartObjects in your Progress applications.

3.1.3 Deciding Which Installation Level To Use

You can install WebSpeed and other Progress products at three levels: Typical, Complete, and Custom. Progress Software Corporation recommends that you perform Typical or Complete Installations. With these methods, you get the components necessary for the products to operate correctly.

If you have limited space on your machines, you might try Custom Installations to reduce the size of the installation. Progress Software Corporation recommends that only expert users perform Custom Installations. If you perform a Custom Installation, make sure you install the components and subcomponents that rely on each other.

Typical Installation

A Typical Installation finds all the components of the product you are installing that are marked mandatory. It then installs only the mandatory subcomponents of those components. Components and subcomponents are marked mandatory if they are required for the product to work in the way most developers use it.

A Typical Installation might not give you all the pieces of the product that you need for your particular purpose. For example, a Typical Installation of ProVision Plus does not install a Development Server.

Complete Installation

A Complete Installation gives you all the components and subcomponents, that you are licensed to use, of the product you are installing. Because it includes the optional as well as mandatory components, a Complete Installation requires the most disk space on your system. A Complete Installation might give you pieces of the product that you do not need.

Custom Installation

You should only consider a Custom Installation when your system resources do not allow a Complete Installation. In a Custom Installation, expert users determine which pieces of the product are installed. Plan Custom Installations carefully and then test them.

A Custom Installation starts by selecting all the mandatory subcomponents of the mandatory components. From there, you can add other components and subcomponents you need. In some cases, you might be able to deselect some of the mandatory components and subcomponents. If you find you need components or subcomponents that you did not install, you can run the Add Component feature to add them.

See Appendix B, "Component Lists For WebSpeed Products," for a list of the components and subcomponents in the Progress products that you might use in a WebSpeed configuration. If you are installing other Progress products, see the "Progress Version 9 Typical and Complete Installation Components" appendix in the *Progress Installation and Configuration Guide Version 9 for Windows* for a complete list of Progress products and their components and subcomponents.

3.1.4 Eliminating NameServer Usage

The WebSpeed Messenger in prior versions of WebSpeed requires connecting to a server instance using a NameServer. In WebSpeed 3.1D, use of the NameServer is optional.

There are several reasons why you may want to eliminate the NameServer from your configuration:

- The NameServer allows for location transparency and load balancing. Simple or static configurations may not require those features.
- The communications protocol employed by the NameServer is the User Datagram Protocol (UDP). Some sites have restrictions that prohibit the use of UDP.

If you choose not to use the NameServer, you will need to do the following:

1. Configure your server in Progress Explorer to indicate that the server should not register with a NameServer. In Progress Explorer, deselect the **Connect Using NameServer** check box in the properties dialog for WebSpeed Messenger. You must also deselect the **Register With NameServer** check box in the properties dialog for the WebSpeed Broker.
2. Connect clients directly to the server by specifying the server's TCP/IP host and port. There is a new connection parameter DirectConnect (`-DirectConnect`) that specifies that no NameServer should be used when the client connects to the server

3.2 Single-machine Configuration

The most basic configuration for WebSpeed is to install the WebSpeed Workshop product on the same machine as your Web server and a data source. The WebSpeed Workshop uses a subset of the Progress development tools focused on developing and testing WebSpeed applications. With the Workshop, you do not have access to the graphical interface design capabilities of the AppBuilder. Figure 3–1 shows the main components installed in a single-machine configuration, as described below.

NOTE: The WebSpeed Broker and the WebSpeed Agents it controls are collectively referred to as the WebSpeed Transaction Server in this figure.

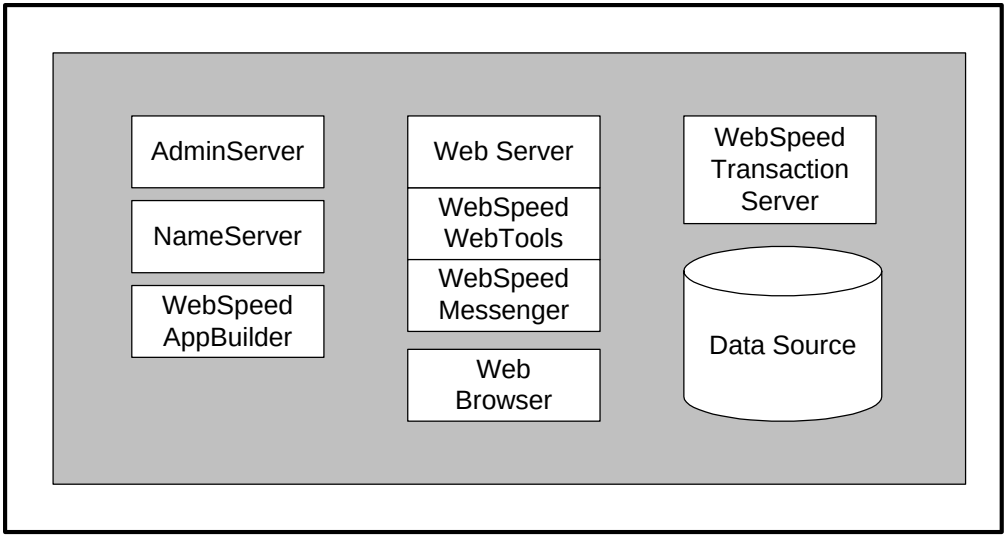


Figure 3–1: Configuration For Single Developer On a Single Machine

The WebSpeed Workshop includes the WebSpeed Development Server, which supports a single developer. The WebSpeed Development Server comes with only two WebSpeed Agents. The WebSpeed Development Server includes the Personal Database. The Personal Database can handle up to three local connections: one for the WebSpeed AppBuilder, one for the Progress Explorer, and one for the applications that you are developing. This is suitable for the needs of a single developer.

As Figure 3–1 shows, the Web server, the WebSpeed WebTools, and the WebSpeed Messenger form an interdependent unit. While the Web server can handle non-WebSpeed traffic in addition to the WebSpeed traffic, the WebSpeed WebTools and Messenger cannot operate independently from the Web server.

To set up this configuration, follow these general steps:

1 ♦ Install a Web browser and a Web server on your machine.

2 ♦ Perform a Complete Installation of the WebSpeed Workshop onto your machine.

You need to perform a Complete Installation because the installation program treats the components that make up the WebSpeed Development Server as optional and does not include them in a Typical Installation.

3.3 Development Network With a Central WebSpeed Machine

A common WebSpeed configuration is a network where a central machine handles most of the WebSpeed components. Individual developers only need the WebSpeed AppBuilder and a Web browser on their workstations. Figure 3–2 shows the main components installed in this configuration, as described below.

NOTE: The WebSpeed Broker and the WebSpeed Agents it controls are collectively referred to as the WebSpeed Transaction Server in this figure.

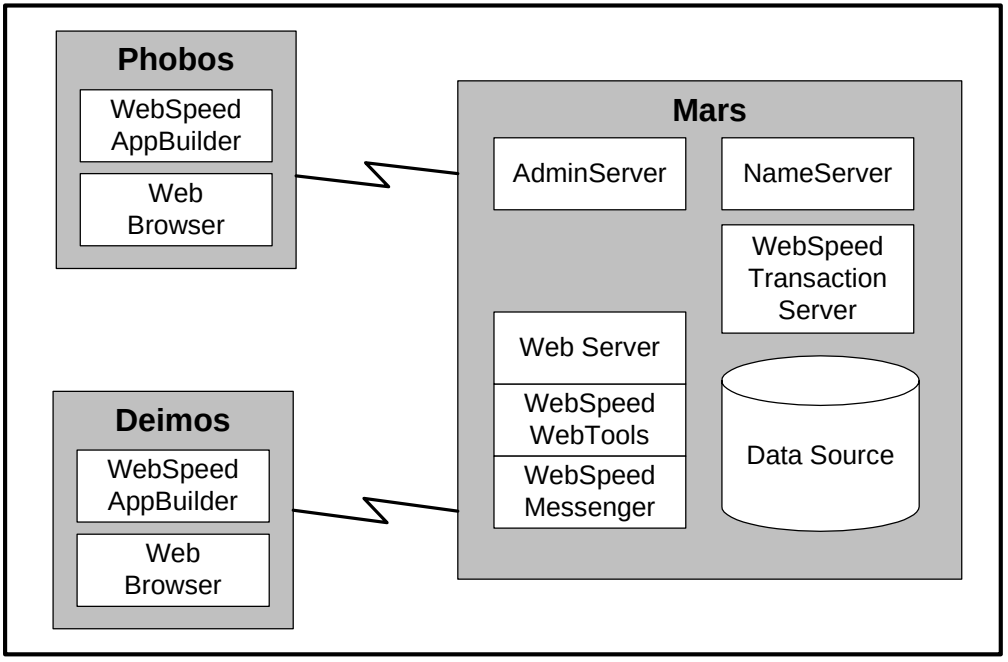


Figure 3–2: Development Network With a Central WebSpeed Machine

This configuration uses the WebSpeed Transaction Server and some data source on the central machine and the WebSpeed Workshop on the workstations. Figure 3–2 shows a central machine, Mars, configured to support two development workstations, Phobos and Deimos.

The WebSpeed Transaction Server supports a team of developers creating WebSpeed applications. You can start up as many WebSpeed Agents as you have licenses for. Depending on the locking strategies for WebSpeed Agents you use in your applications, any given Agent might serve the needs of several developers or only one developer. Because the WebSpeed Transaction Server does not come with a database, you need to install either the Workgroup Database or the appropriate DataServer to connect to your non-Progress database.

To set up this configuration, follow these general steps:

1 ♦ Install a Web server on Mars.

2 ♦ Perform a Typical Installation of the WebSpeed Transaction Server on Mars.

3 ♦ Install the appropriate components for your intended data source on Mars:

- If you use the Progress Database, perform a Typical Installation of the Workgroup Database.
- If you connect to a non-Progress data source, install the appropriate Progress DataServers for your data source. See the appropriate Progress DataServer guide for details.

4 ♦ Perform Typical Installations of the WebSpeed Workshop on Phobos and Deimos.

Beyond the Basic Installation

If you are going to use a non-Progress database, you need to install the appropriate DataServers. Perform Complete Installations, rather than Typical Installations, of the WebSpeed Transaction Server on the Mars server and the WebSpeed Workshop on Phobos and Deimos.

3.4 Development Network With a Dedicated Web Server

Another common configuration is a network made up of workstations; a machine that handles the Web server, WebSpeed WebTools, and WebSpeed Messenger; and a machine that handles the WebSpeed Transaction Server, the applications, and data source. Figure 3–3 shows the main components installed in this configuration, as described below.

NOTE: The WebSpeed Broker and the WebSpeed Agents it controls are collectively referred to as the WebSpeed Transaction Server in this figure.

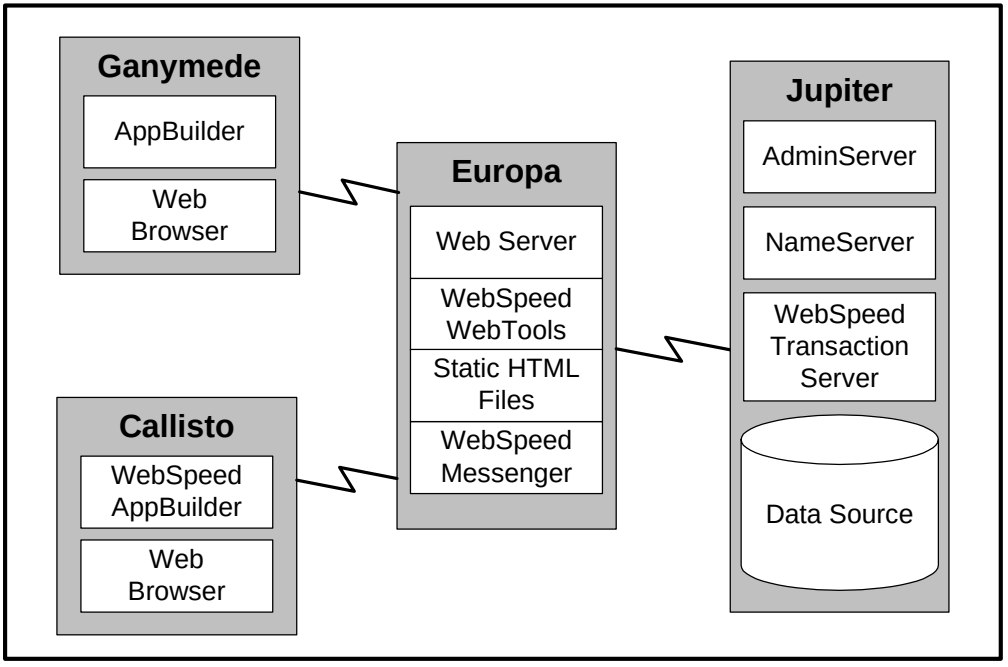


Figure 3–3: Development Network With a Dedicated Web Server

This configuration has the WebSpeed Messenger installed on the Web server's machine, Europa. The host machine, Jupiter, uses the WebSpeed Transaction Server. Because the WebSpeed Transaction Server does not come with a database, you must install the Workgroup Database or the appropriate DataServer to connect to your non-Progress database.

One of the workstations, Ganymede, has ProVision Plus installed to gain access to the full capabilities of the AppBuilder. The developer using Ganymede can alternate between developing Web applications and non-Web-based applications. The other workstation, Callisto, has the WebSpeed Workshop installed. The developer using Callisto can only develop Web applications with the WebSpeed AppBuilder.

To set up this configuration, follow these general steps:

1 ♦ Install a Web server on Europa.

2 ♦ Perform a Typical Installation of the WebSpeed Transaction Server on Jupiter.

3 ♦ Install the appropriate components for your intended data source on Jupiter:

- If you use the Progress Database, perform a Typical Installation of the Workgroup Database.
- If you connect to a non-Progress data source, install the appropriate Progress DataServers for your data source. See the appropriate Progress DataServer guide for details.

4 ♦ Perform a Typical Installation of the WebSpeed Messenger on Europa.

5 ♦ Copy the static HTML and GIF (`.gif`) files from Jupiter's `DLC/progress/webspeed` directory to your Web server's `/docroot/webspeed` directory on Europa. These are the files for the WebSpeed WebTools that must reside with your Web server.

6 ♦ Perform a Typical Installation of ProVision Plus on the Ganymede workstation.

7 ♦ Perform a Typical Installation of the WebSpeed WorkShop on the Callisto workstation.

Beyond the Basic Installation

If you need access to any of the DataServers for your applications, you can add them by performing a Complete Installation of the WebSpeed Transaction Server on Jupiter, ProVision Plus on Ganymede, and the WebSpeed Workshop on Callisto.

3.5 Multiple LAN Development Environment

Some development environments include several local area networks (LANs). For example, a company's intranet might include two LANs—a Sales LAN and a Manufacturing LAN.

The common Progress administration architecture includes a method for connecting NameServers to other NameServers. When a NameServer receives a request for an application service, it checks to see whether any of the WebSpeed Brokers registered with it supports that application service. If none of its registered brokers supports that application service, the NameServer passes the request to other NameServers that it knows about.

Using this capability, you can link Progress installations across the LANs that make up a wide area network. Figure 3–4 shows the products installed on a multiple LAN configuration.

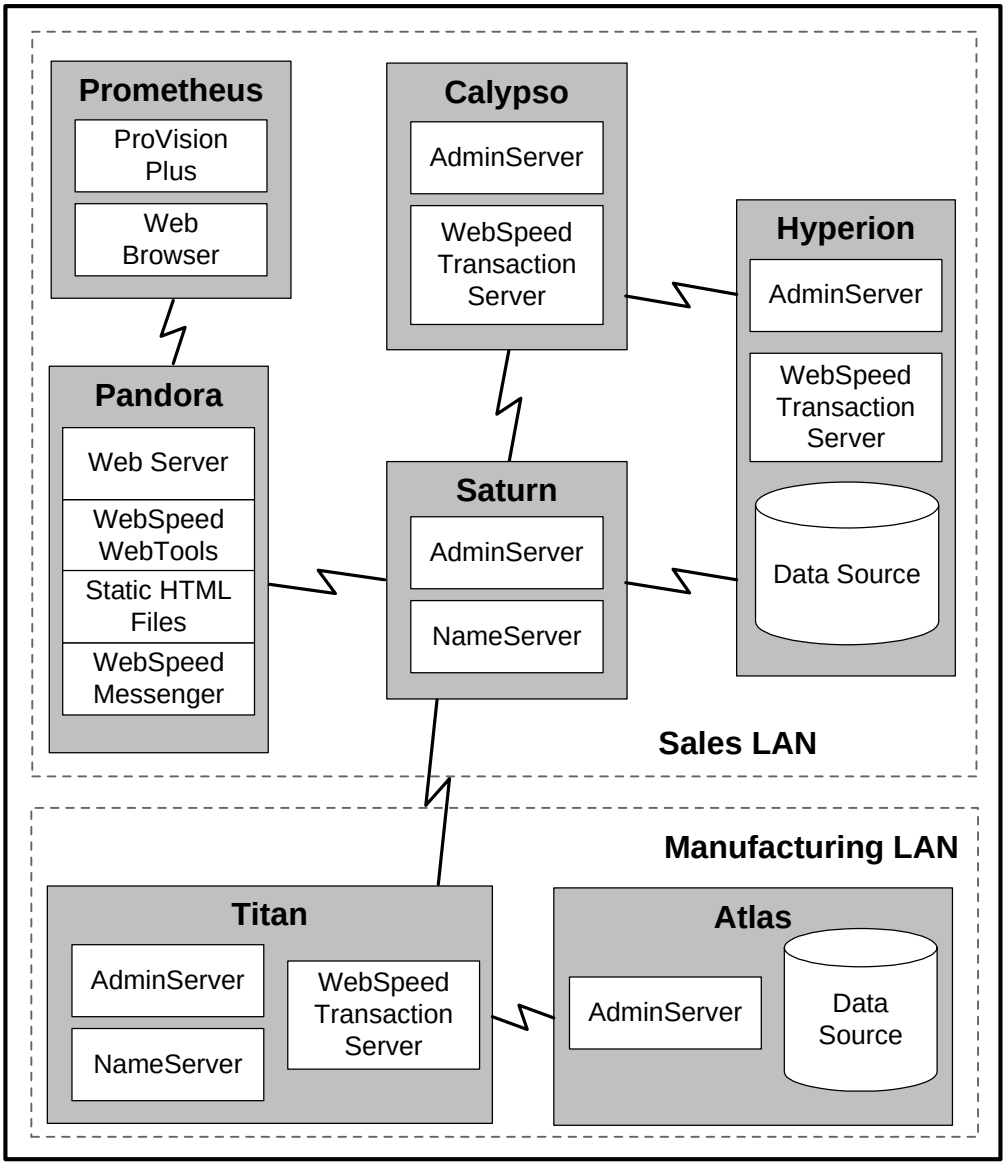


Figure 3–4: Multiple LAN Development Environment

In this configuration, the NameServer machine, Saturn, is the central communication point that routes requests to the WebSpeed Transaction Servers on both LANs. Requests start from workstations, like Prometheus (which has ProVision Plus installed on it). A workstation passes the request to the Web server machine, Pandora, which has the WebSpeed Messenger installed. The Messenger asks the NameServer on Saturn to find a WebSpeed Transaction Server that can handle the request, by specifying an application service with the WService parameter.

Calypso and Hyperion both have the Transaction Server installed. The Transaction Servers are registered with Saturn's NameServer. Saturn's NameServer checks its list of registered Transaction Servers to see whether one supports the application service that can handle the request from Pandora's Messenger. If one of them does, the NameServer passes its location back to the Messenger and the request is completed as shown in the flow diagram in the "How WebSpeed Components Work Together" section of Chapter 2, "Overview Of the WebSpeed Architecture."

The Transaction Server installed on Titan includes a NameServer. You can register this NameServer with Saturn's NameServer. If the WebSpeed Brokers on Calypso or Hyperion do not support the correct application service, Saturn's NameServer passes the request to the NameServer on Titan to see whether the Broker registered with it supports that application service. If the Broker on Titan does support the application service, Titan's NameServer passes its location to Pandora's Messenger. Then the request is completed as normal.

For more information on how this works, see the "Using NameServer Neighbors" section of the "Configuring Progress Unified Broker Products" chapter in the *Progress Installation and Configuration Guide Version 9 for Windows*.

Note that there are only two machines with data sources in this configuration—Hyperion, on the NT side, and Atlas, on the UNIX side. Because the WebSpeed Transaction Server does not come with a database, you must install the Enterprise Database or the appropriate DataServer to connect to your non-Progress database. The WebSpeed Agents on any of the Transaction Server machines in the configuration can access either data source.

To set up this configuration, follow these general steps:

1 ♦ Install a Web server on Pandora.

2 ♦ Perform a Typical Installation of the NameServer on Saturn.

3 ♦ Perform Typical Installations of the WebSpeed Transaction Server on Calypso, Hyperion, and Titan.

4 ♦ Perform a Typical Installation of the WebSpeed Messenger on Pandora.

5 ♦ Copy the static HTML and GIF files of the `DLC/progress/webspeed` directory on one of the Transaction Server machines to your Web server's `/docroot/webspeed` directory on Pandora. These are the files for the WebSpeed WebTools that must reside with your Web server.

6 ♦ Perform Typical Installations of ProVision Plus or the WebSpeed Workshop on Prometheus and your other workstations, according to your development needs.

7 ♦ Install the appropriate data source components on Hyperion and Atlas. This depends on what you use as your data source:

- If you use the Progress Database, perform a Typical Installation of the Enterprise Database.
- If you connect to a non-Progress data source, install the appropriate Progress DataServers. See the appropriate Progress DataServer guide for details.

Beyond the Basic Installation

When you expect heavy loads on your network or want to increase its fault tolerance, you might want to use the load balancing capability of the Enterprise NameServer. If you deploy several WebSpeed Transaction Servers that support the same application service, you can use load balancing to control how the load is spread over them. See the "Understanding the NameServer's Load Balancing Option" section in Chapter 7, "Configuring WebSpeed On Windows," and the "Configuring Progress Unified Broker Products" chapter in the *Progress Installation and Configuration Guide Version 9 for Windows*.

If you need access to any of the DataServers for your applications, you can add them by performing a Complete Installation of the WebSpeed Transaction Server on those machines and ProVision Plus on your workstations.

3.6 Deployment Configuration With a Dedicated Web Server

When you deploy your applications, you do not need all the components that usually exist in a development configuration. A deployment configuration might look like the one shown in Figure 3–5.

NOTE: The WebSpeed Broker and the WebSpeed Agents it controls are collectively referred to as the WebSpeed Transaction Server in this figure.

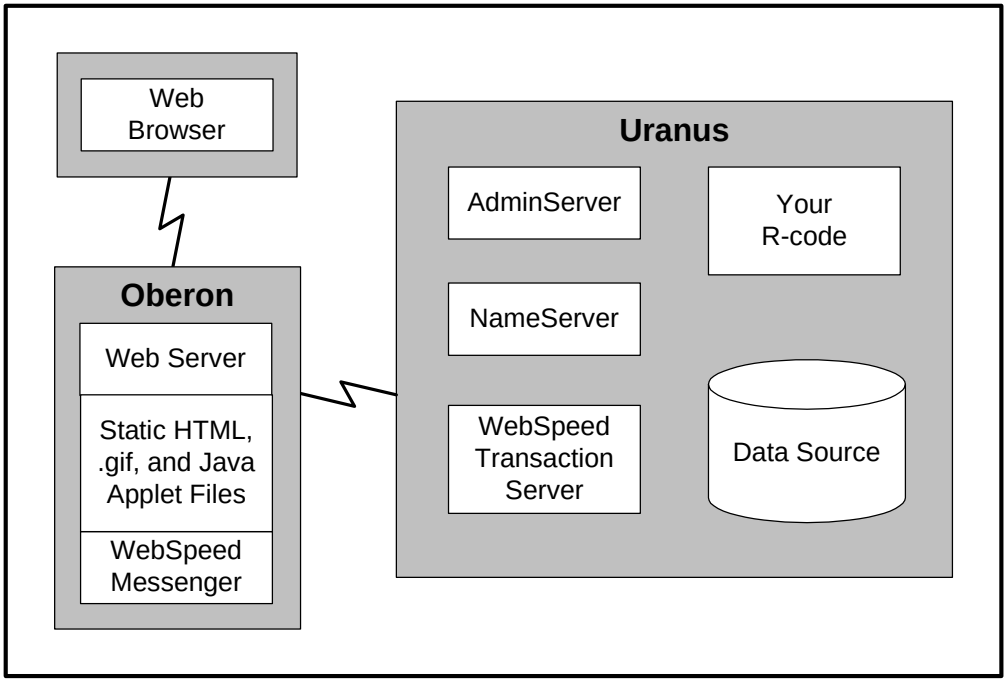


Figure 3–5: Deployment Configuration With a Dedicated Web Server

This configuration has the WebSpeed Messenger product installed on the Web server's machine, Oberon. The main machine, Uranus, has the WebSpeed Enterprise Transaction Server and some data source. The end users only need a Web browser to link up to your applications. They do not need any Progress product installed on their machines.

To set up this configuration, follow these general steps:

1 ♦ Install a Web server on Oberon.

2 ♦ Perform a Typical Installation of the WebSpeed Messenger on Oberon.

NOTE: Since the WebSpeed WebTools are only meant for developers, do not copy the files from the `DLC/progress/webspeed` directory on Uranus into the Web server's `/docroot` directory in this configuration.

3 ♦ Perform a Typical Installation of the WebSpeed Enterprise Transaction Server on Uranus.

4 ♦ Install the appropriate components for your intended data source on Uranus:

- If you use the Progress Database, perform a Typical Installation of the Enterprise Database.
- If you connect to a non-Progress data source, install the appropriate Progress DataServers for your data source. See the appropriate Progress DataServer guide for details.

5 ♦ Install your r-code on Uranus and include its location in your PROPATH.

6 ♦ Copy any Java applet, static HTML, and GIF files needed for your applications to the Web server's `/docroot/webspeed` directory on Oberon.

Beyond the Basic Installation

As described, this configuration uses the components for a heavy load system. If you expect a light load on the network, you might consider installing the WebSpeed Transaction Server and Workgroup Database instead of the Enterprise Transaction Server and Enterprise Database. The WebSpeed Transaction Server can handle up to 50 WebSpeed Agents. The Workgroup Database can handle up to 65 connections.

If you need access to any of the DataServers for your applications, you can add them by performing a Complete Installation of the WebSpeed Transaction Server on Uranus.

3.7 Development and Deployment In a Shared Configuration

When a company develops its own applications, it can have networks that are used for development, testing, and deployment simultaneously. Figure 3–6 shows this type of configuration.

NOTE: The WebSpeed Broker and the WebSpeed Agents it controls are collectively referred to as the WebSpeed Transaction Server in this figure.

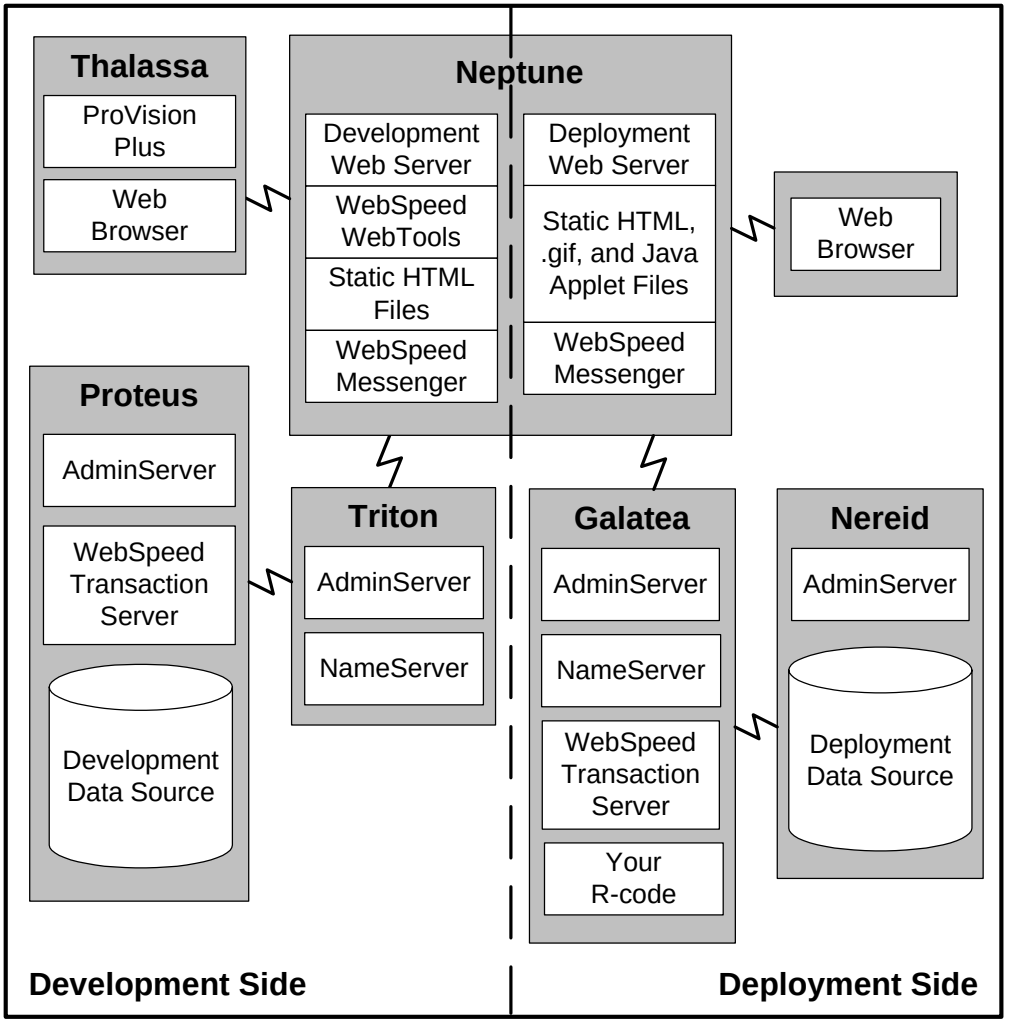


Figure 3–6: Development and Deployment In a Shared Configuration

In this configuration, the dedicated Web server machine, Neptune, has two Web servers installed. One Web server handles the development workload, and the other handles the deployment workload. This allows the developers to experiment with the environment without bringing down the deployment side. Each of the Web servers requires its own WebSpeed Messenger, but only the development Web server needs the WebSpeed WebTools.

On the development side, the Messenger routes requests to the NameServer on Triton. Triton's NameServer only has the WebSpeed Transaction Servers on Proteus registered, so it looks there for a Transaction Server to handle the request. Proteus has the WebSpeed Enterprise Transaction Server and either the Enterprise Database or a non-Progress data source installed on it.

On the deployment side, that Messenger routes requests to the NameServer on Galatea. Galatea has the WebSpeed Enterprise Transaction Server installed. The deployment side uses a dedicated machine, Nereid, to hold its data source (that is the Enterprise Database or a non-Progress data source).

To set up this configuration, follow these general steps:

1 ♦ Install two copies of your Web server on Neptune.

2 ♦ Perform a Typical Installation of the Enterprise NameServer on Triton.

3 ♦ Perform a Typical Installation of the WebSpeed Enterprise Transaction Server on Proteus.

4 ♦ Perform a Typical Installation of the WebSpeed Messenger on Neptune using the development Web server's Document Root directory and scripts path.

5 ♦ Copy the static HTML and GIF files of the `DLC/progress/webspeed` directory on Proteus to the Development Web server's `/docroot/webspeed` directory on Neptune. These are the files for the WebSpeed WebTools that must reside with your Web server.

6 ♦ Configure the Development Web server's WebSpeed Messenger on Neptune to pass requests to Triton's NameServer.

7 ♦ Perform Typical Installations of ProVision Plus or the WebSpeed Workshop on Thalassa and your other workstations, according to your development needs.

8 ♦ Perform a Typical Installation of the WebSpeed Messenger on Neptune using the deployment Web server's Document Root directory and scripts path.

9 ♦ Perform a Typical Installation of the WebSpeed Enterprise Transaction Server on Galatea.

10 ♦ Configure the Deployment Web server's WebSpeed Messenger on Neptune to pass requests to Galatea's NameServer.

11 ♦ Install the appropriate data source components on Proteus and Nereid. This depends on what you are planning to use as your data source:

- If you use the Progress Database, perform a Typical Installation of the Enterprise Database.
- If you connect to a non-Progress data source, install the appropriate Progress DataServers. See the appropriate Progress DataServer guide for details.

12 ♦ Install your r-code on Galatea.

13 ♦ Copy any Java applet, static HTML, and GIF files that your applications need to the deployment Web server's `/docroot/webspeed` directory on Neptune.

Beyond the Basic Installation

As described, this configuration uses the components for a heavy load system. If you expect a light load on the network, you might consider installing the WebSpeed Transaction Server and Workgroup Database instead of the Enterprise Transaction Server and Enterprise Database. The WebSpeed Transaction Server can handle up to 50 WebSpeed Agents. The Workgroup Database can handle up to 65 connections.

You can expand the capacity of either side of this configuration by adding new server machines with the same products installed. Register the WebSpeed Transaction Servers on the new machines with the NameServer that is handling the Transaction Servers for that side.

When you expect heavy loads on your network or want to increase its fault tolerance, you might want to use the load balancing capability of the Enterprise NameServer. If you deploy several WebSpeed Transaction Servers that support the same application service, you can use load balancing to control how the load is spread over them.

If you need access to any of the DataServers for your applications, you can add them by performing Complete Installations of the WebSpeed Enterprise Transaction Server on the appropriate machines and ProVision Plus on your workstations.

4

Before You Install WebSpeed

Installing and configuring WebSpeed is easier when you make the proper preparations. This chapter describes what to do before you install WebSpeed.

This chapter covers the following topics:

- Preparing Your Web Server
- Planning Your WebSpeed Installation
- Matching Your Web Server and WebSpeed Messenger
- Preinstallation Tasks

4.1 Preparing Your Web Server

WebSpeed relies on your Web server to receive queries from and send replies to HTML clients. Before installing WebSpeed, your Web server must be installed and operating properly. The following sections discuss the things you should know about your Web server, present some simple diagnostic tests, and show how to create virtual directories and additional document directories for your Web server.

4.1.1 Understanding Your Web Server

The Web server is the backbone of any Web application. The WebSpeed Transaction Server is the engine that drives your Web applications, but it relies on the Web server to receive and pass on requests that users send across the Internet through their browsers. The WebSpeed Messenger resides on the machine where your Web server runs, ready to react to incoming requests for data that a WebSpeed application can fill.

You need a working knowledge of Web servers to install and configure WebSpeed successfully. Table 4–1 summarizes what you must know about your Web server before you can start installing and configuring WebSpeed.

Table 4–1: About Your Web Server *(1 of 2)*

Information	Comments
Type of Web Server	You must know whether your Web server is IIS, Netscape, or CGI compatible. Almost all Web servers support CGI. If you plan to use the ISAPI Messenger, your Web server must be IIS compatible. If you plan to use the NSAPI Messenger, your Web Server must be Netscape compatible. Otherwise, use the CGI Messenger.
Host Name	You must know the name by which your network recognizes the Web server machine. Optionally, you can use the Web server's domain name, that is, the name by which the Internet recognizes it.

Table 4–1: About Your Web Server *(2 of 2)*

Information	Comments
Document Root Directory	The Web server must be able to locate the static files that the WebSpeed Workshop or any WebSpeed application requires. To ensure that the Web server can locate the files, you can physically place the files in the Web server's document root directory or you can configure the Web server so that its document root directory points to the actual location of the files.
Script Directory	You must know in which directory your Web server expects to find the executables and scripts that it uses to process incoming Web requests. For Web servers on Windows NT, this directory is a virtual directory that you specify when you configure the Web server. The WebSpeed Messenger executable must reside in this directory (except for the NSAPI Messenger, which resides in *install-path*\bin). On UNIX, copy the shell script, wspd_cgi.sh, into your Web server's scripts directory. This shell script invokes the WebSpeed Messenger executable.
Starting and Stopping	You must know how to start and stop your Web server. After installing the Messenger, you must restart the ISAPI or NSAPI Web server. For Netscape Web servers, you must also apply the configuration changes you make. You do not have to restart a CGI Web server.

4.1.2 Testing Your Network With the ping Command

To have your Web server work, you must have a properly functioning network. Your Web server has to be connected to a network to receive requests from clients on other machines. You can test your network with the ping command. The ping command is available on any machine that is connected to a network using the TCP/IP protocol. It sends TCP/IP test packets to the computer you specify and verifies the packets it receives in reply.

When you use the ping command, send the first ping to "localhost." This verifies that your local network connection is working properly. After checking that your local network connection is working, you can use the ping command to verify that remote machines are connected to your network.

You can find the specifics for using the ping command on your system through the Help system of your Windows machine or through the man page for the ping command on your UNIX machine.

4.1.3 Reaching Your Web Server's Home Page

Even if you have not created a personal or company home page on your Web server, every Web server has a default home page. You can check that your Web server is working by connecting to its home page from your Web browser. The URL for its default home page is `http://`*`host name`*.

4.1.4 Creating Directories On Your Web Server

You might have to create additional directories on your Web server to properly install and configure WebSpeed. Read your Web server's documentation to learn how it manages its directory structure. Before you install WebSpeed, you should understand how to do the following tasks:

- Create a virtual directory for your Web server.
- Create a scripts directory for your Web server.
- Create additional document directories for your Web server.

4.2 Planning Your WebSpeed Installation

See Chapter 3, "WebSpeed Configurations," for information on some of the many ways you can set up your WebSpeed environment. How you install WebSpeed depends on whether you are setting up a development environment or a deployment environment. It also depends on how you distribute the components that make up WebSpeed to optimize your hardware and network resources.

Before beginning the installation, carefully plan how you want the final environment configured. Pay special attention to which components have to reside on the same physical machine.

NOTE: If you currently have earlier versions of WebSpeed installed, see Chapter 12, "Upgrading WebSpeed On Windows," or Chapter 13, "Upgrading WebSpeed On UNIX," for coexistence and migration information you should consider in planning your new installation of WebSpeed.

4.2.1 Connection Limits

One of the factors you should consider in planning your installation is how heavy a load you expect. The number of connections your environment can support depends on which products you install. Progress Software Corporation offers three levels of database licensing: Personal, Workgroup, and Enterprise. There are also four different Transaction Server products you might use in your environment.

Table 4–2 lists the number of connections you can have with each level of license for the Progress database.

Table 4–2: Database Connection Limits

Database License	Maximum Connections
Personal Database	3 (local only)
Workgroup Database	30 (approximate practical limit) 49 (absolute maximum)
Enterprise Database	10,000

Table 4–3 lists the number of WebSpeed Agents each of the Transaction Servers can support.

Table 4–3: WebSpeed Agent Limits For Transaction Servers

Transaction Server	Maximum WebSpeed Agents
ProVision Plus Development Server	2
WebSpeed Development Server	2
WebSpeed Transaction Server	5 or 25
WebSpeed Enterprise Transaction Server	50 or 250

4.3 Matching Your Web Server and WebSpeed Messenger

If you plan to run the WebSpeed Transaction Server and your Web server on separate operating systems, you must obtain the WebSpeed Messenger for your Web server's operating system. The Messenger must also be compatible with the interface standard that your Web server supports. Almost all Web servers support the CGI interface standard, but the CGI Messenger tends to have the slowest response time. You can download all the Messengers from the Download area on Progress Technical Support's Web site, `http://www.progress.com/services/support/`.

Table 4–4 lists the most common Web servers and the appropriate WebSpeed Messengers.

Table 4–4: Web Server and Messenger Types

Web Server	Messenger
Microsoft Internet Information Server, Versions 3.x and 4.x	CGI Messenger (`CGIIP.EXE`) ISAPI Messenger `WSISA.DLL`
Netscape Enterprise or Fast Track Server, Version 3.x	CGI Messenger on UNIX (`cgiip`) CGI Messenger on NT (`CGIIP.EXE`) NSAPI Messenger `WSNSA.DLL`
Apache	CGI Messenger on UNIX (`cgiip`) CGI Messenger on NT (`CGIIP.EXE`)

The Messenger product consists of the executables listed in Table 4–4 and additional configuration files, utility scripts, and messages files (PROMSGS).

4.4 Preinstallation Tasks

After you have planned your WebSpeed installation, complete the preinstallation tasks described in Table 4–5 before starting your WebSpeed installation.

Table 4–5: Preinstallation Tasks *(1 of 2)*

Task	Description
Read	Read the hard-copy *Release Notes*. The *Release Notes* might replace or supplement the installation instructions in this guide.
Shut down	On Windows systems, 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 WebSpeed.
Preinstall	Install a Web Server product and configure a Web Server directory on a machine where you want these products to reside. Install a Web browser on machines where you plan to install an AppBuilder. Install any required Java components. See the "Java Requirements" section of Chapter 1, "Installation Requirements," for detailed information.
Check	Check that you have enough system resources for the installation and that all environment variables have been properly set. On UNIX, make sure that the ULIMIT on your system is set to at least 8MB. For specific instructions on setting the ULIMIT, consult its man page.
Delete (NT only)	If you have a previous version of WebSpeed and the PROMSGS environment variable is set in the Control Panel on Windows NT, you must delete the PROMSGS environment variable before installing WebSpeed. If PROMSGS points to an old or nonexistent PROMSGS file, the InstallShield Utility will not write all the necessary data to the Windows registry.
Copy	If you have an existing WebSpeed installation, make a copy of the Document Root directory and the Scripts directory before you install this new version of WebSpeed. The new installation overwrites the WebSpeed files in these directories. The working directory is unchanged when you install a new version of WebSpeed. It is possible, however, that the WebSpeed log files might be overwritten.

Table 4–5: Preinstallation Tasks *(2 of 2)*

Task	Description
Convert	If you are upgrading a Version 8.x Progress database, truncate the before-image (.bi) file using the PROUTIL TRUNCATE BI utility and back up your Version 8.x Progress database using the PROBKUP utility. For more information on converting databases, see the "Creating and Deleting Databases" chapter in the *Progress Database Administration Guide and Reference*.
Note	Note the following information about your Web server: • **Web server API type**—ISAPI, NSAPI, or CGI. Your Web server can be running on any platform. • **Web server Document Root directory**—You can choose to have the setup program install the HTML static files for WebSpeed in this directory, if you install your Transaction Server on the same machine as your Web server. • **Web server scripts path**—This depends on whether your Web server is on a Windows NT machine or a UNIX machine: – For a Windows NT server, the setup program installs the CGI and ISAPI Messenger executables into the scripts directory of your Web server. It installs the NSAPI Messenger executable into the *install-path*\bin directory. – For a UNIX server, the installation utility installs the CGI Messenger script into the directory that your Web server uses to locate scripts and executables. It installs the NSAPI and CGI Messenger executables into the *install-path*/bin directory.

CAUTION: The WebSpeed Development Server and WebSpeed Workshop products include the Progress Version 9 Personal Database. If you are using Progress Version 8 single-volume databases, 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 before-image 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 WebSpeed and avoid erasing your Progress Version 8 files.

5

Installing WebSpeed On Windows

This chapter contains step-by-step instructions for installing WebSpeed on supported Windows platforms. The chapter contains information on the following:

- Beginning the Installation
- Performing a Typical Or a Complete Installation
- Performing a Custom Installation
- Installing Additional Products
- Network Installations
- The InstallShield Silent (Batch Mode) Utility
- Uninstalling WebSpeed

5.1 Beginning the Installation

How you install WebSpeed depends on what products, components, and subcomponents you choose to install. You can tailor your installation so that only those products and components essential to your business operation are installed.

As you proceed through the installation, you will be asked to choose one of the following installation options:

- **Typical**—Installs the most commonly used WebSpeed components and subcomponents. This is the default installation.
- **Complete**—Installs all WebSpeed components and subcomponents.
- **Custom**—Installs only the WebSpeed components and subcomponents that you choose.

Read the system requirements information in Chapter 1, "Installation Requirements," before performing the installation.

Then follow these steps to begin installing WebSpeed:

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

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

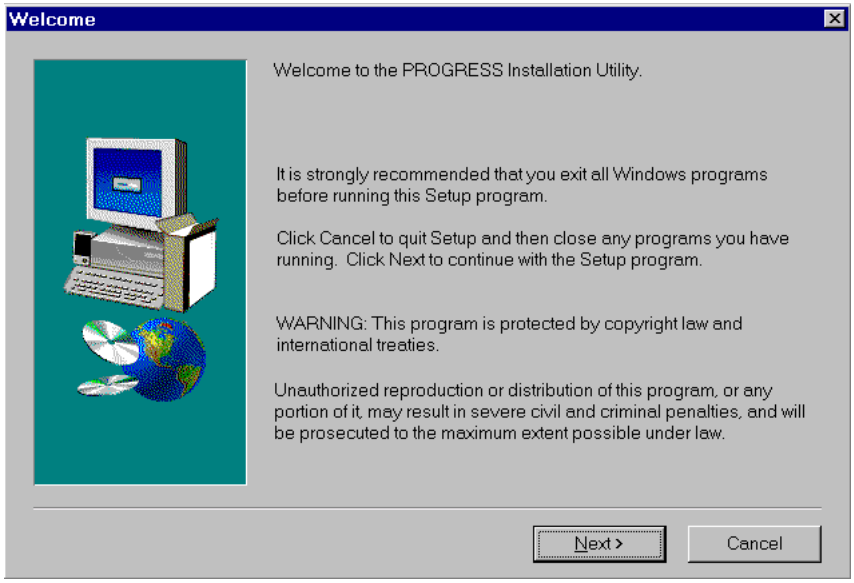
NOTE: Do not run other applications during the installation process.

3 ♦ Insert the installation CD into your CD-ROM drive. Choose Run, and type this command in the Run dialog box:

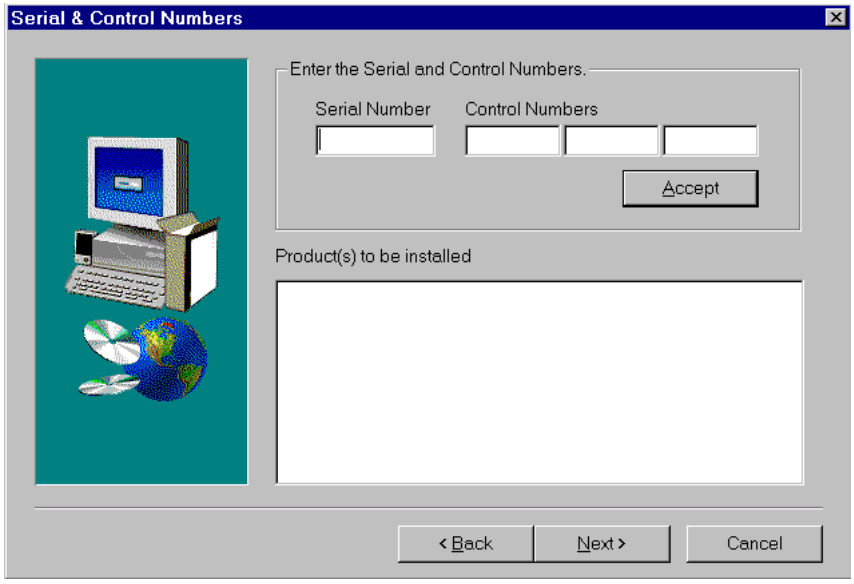
```
drive:\setup.exe
```

where *drive* is the name of your CD-ROM drive.

The Welcome dialog box appears:



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



5 ♦ Type in the nine-digit serial number and control numbers from your *License Addendum*. Note that the cursor advances to the next field automatically when you enter the right number of characters.

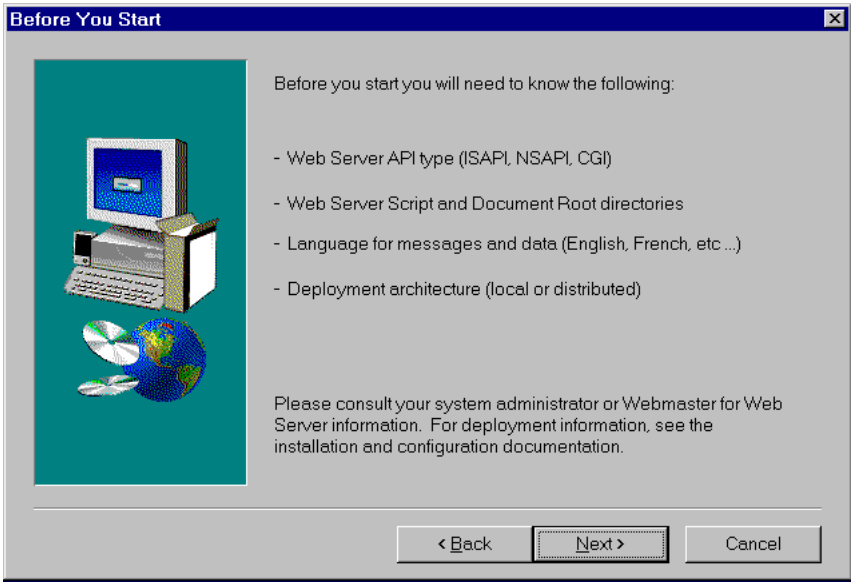
This information is not case sensitive. The installation process automatically converts all letters to uppercase.

You can specify more than one product to install. Type in the serial number and control codes for each product that is included in your *License Addendum*. Choose Accept.

The product name appears in the Product(s) to be installed panel.

6 ♦ Once you have entered all the products you want to install, choose Next to continue.

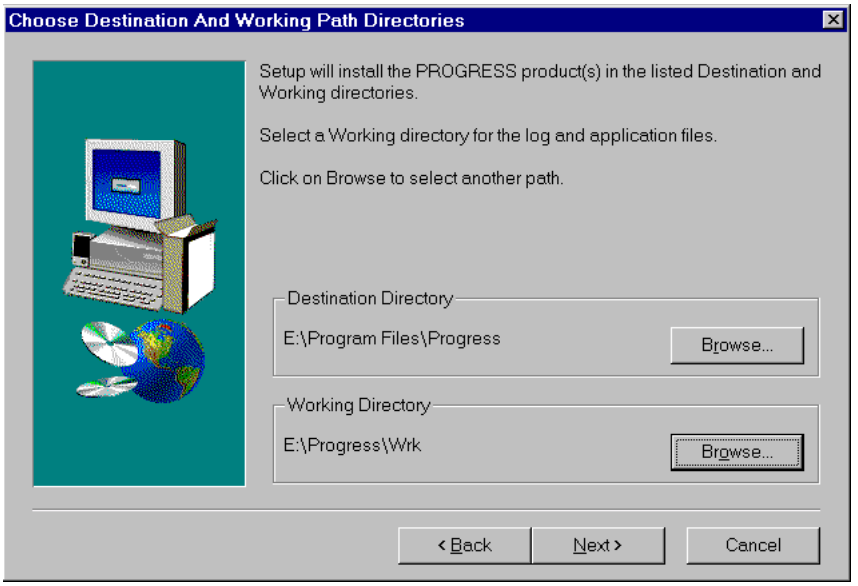
The Before You Start dialog box appears:



7 ♦ Before you continue with the installation, make sure that you know the following:

- **Web Server type**—You must know which type of Web server you have installed: Microsoft IIS, Netscape, or CGI-compatible. You choose which WebSpeed Messenger to install based on the kind of Web Server you have installed.
- **Web Server Script directory**—You must know the absolute pathname of the directory that your Web server uses to store scripts. The setup program must install the WebSpeed ISAPI and CGI Messenger executables into that directory. It installs the NSAPI Messenger script into *install-path*\bin.
- **Web Server Document Root directory**—You must know the absolute pathname of the directory that your Web server uses to store static HTML files. The setup program can copy the WebSpeed static HTML files, such as documentation files and the files required by Workshop, into that directory. The static files reside in *install-path*\webspeed. Optionally, you can reconfigure your Web server by creating an additional document directory that points to where you installed the WebSpeed static files.
- **Language**—You must decide which language you want WebSpeed to use when displaying messages and handling data.
- **Deployment architecture**—You must decide whether you want to install all WebSpeed components on one machine or distribute them across two or more machines.

8 ♦ Choose the Next button. The Choose Destination And Working Path Directories dialog box appears:



9 ♦ Enter your destination path or accept the default directory in which to install WebSpeed (`c:\Program Files\PROGRESS`).

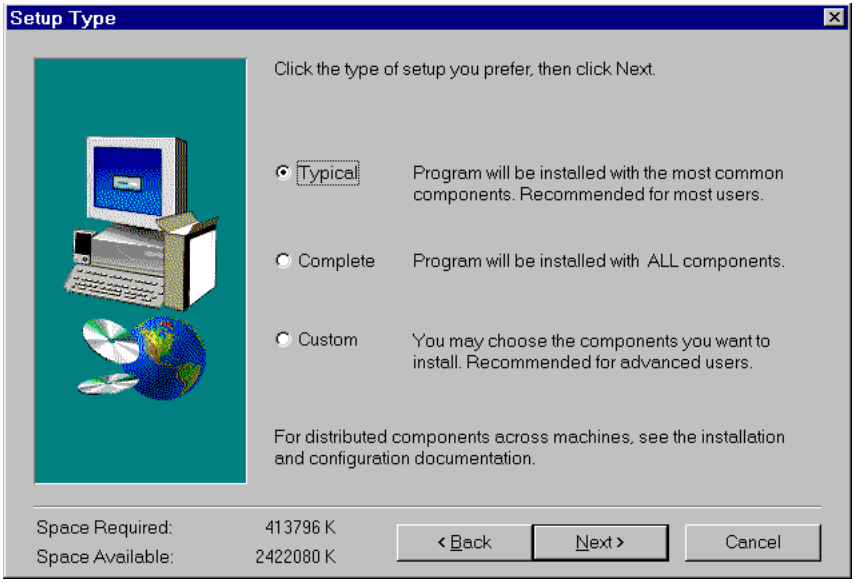
CAUTION: Do not make your Progress working directory a subdirectory under the Progress installation path. If you uninstall Progress, the working directory is deleted along with the destination directory.

10 ♦ Enter a working directory or accept the default (`C:\PROGRESS\WRK`). Your applications, databases, and log files reside in this directory. This directory also serves as the default working directory for WebSpeed Agents.

CAUTION: If you have WebSpeed or Progress products installed, do **not** enter the pathname where these products reside. You must specify a different destination path for WebSpeed Version 3.1. This ensures that you do not overwrite any of your existing WebSpeed or Progress product directories and files.

If you want to add additional Progress products to the most recently installed version, the install program assumes that you are adding products to that install location.

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



12 ♦ Choose the type of installation you want to perform—Typical, Complete, or Custom.

5.2 Performing a Typical Or a Complete Installation

With WebSpeed Version 3.1, 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:

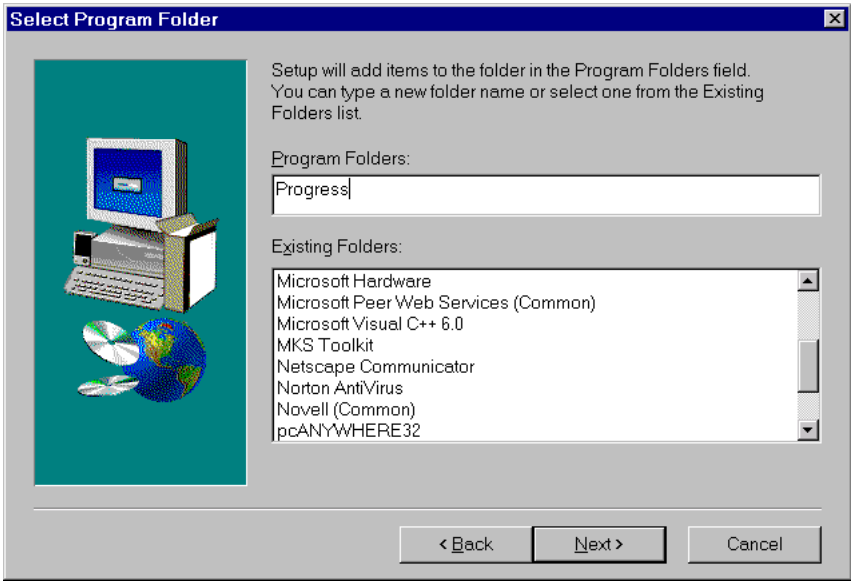
- **Typical**—Perform a typical installation when you want to install the most commonly used WebSpeed components. A typical installation saves disk space by installing only the most widely used WebSpeed components and subcomponents.
- **Complete**—Perform a complete installation when you want to install all WebSpeed components and subcomponents. For example, choose this option if you intend to install the WebSpeed or ProVision Plus development servers.

NOTE: If you choose a Custom installation, see the "Performing a Custom Installation" section.

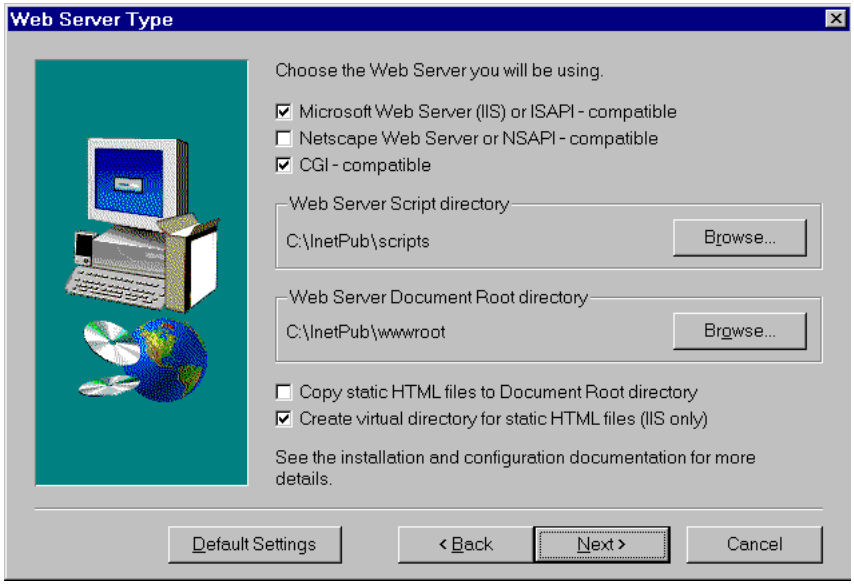
Follow these steps to perform either a Typical or Complete installation:

1 ♦ Complete the steps in the "Beginning the Installation" 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 ♦ Select the program folder that will contain the WebSpeed files, type a new program folder name, or accept the default, and choose Next to continue. The Web Server Type dialog box appears:

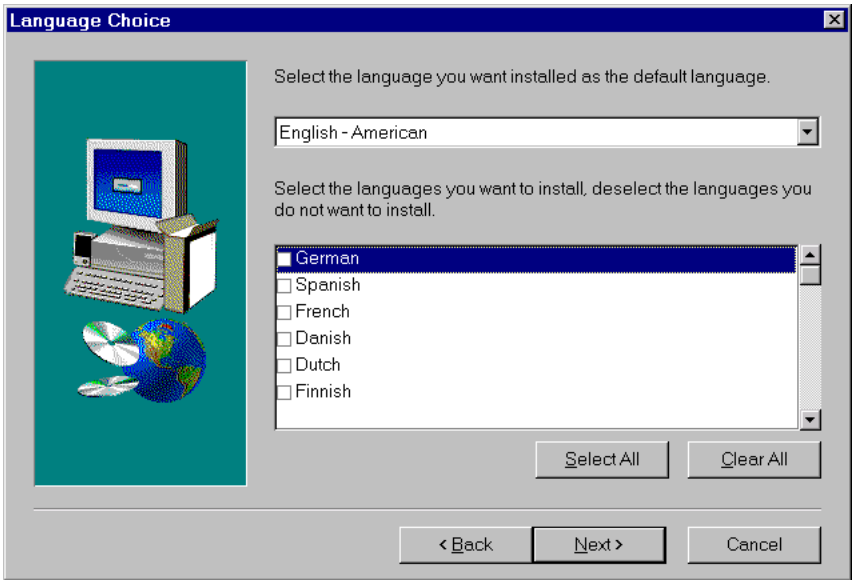


Select the Web Server types for which you want to install Messengers and provide the pathnames for the Web server's Script and Document Root directories.

4 ♦ Choose whether you want the WebSpeed static files copied to your Web server's document root directory or, if you are using the Microsoft IIS, have the setup program create a virtual directory for the static files. The static files physically reside in *install-path*\webspeed. See the "Beginning the Installation" section for more information about static files.

To reset the dialog box settings to those chosen in a previous installation, choose Default Settings.

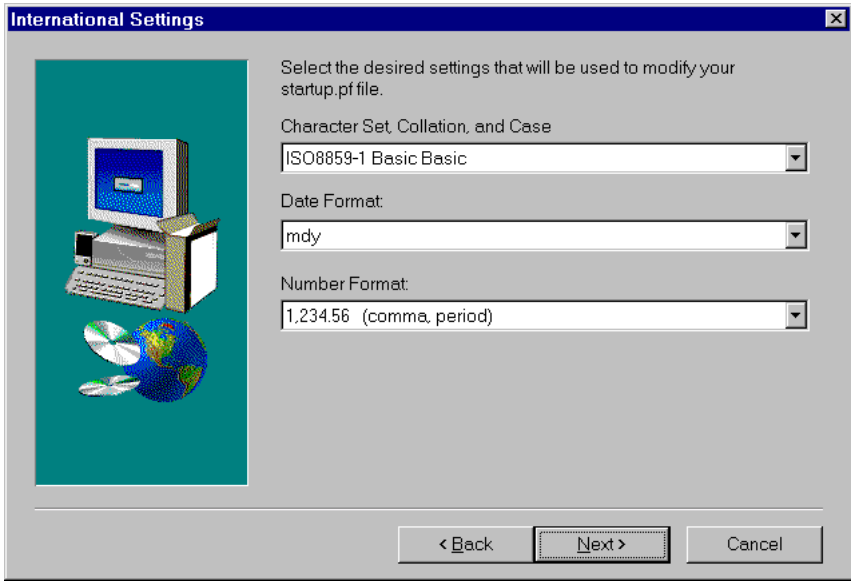
5 ♦ Choose Next to continue. The Language Choice dialog box appears:



6 ♦ If you do not want American English as your default language, select the default language that WebSpeed uses for messages and handling data from the drop-down list.

Select other languages for which you want to install error message files. You can configure WebSpeed Agents to display messages in languages other than the default language.

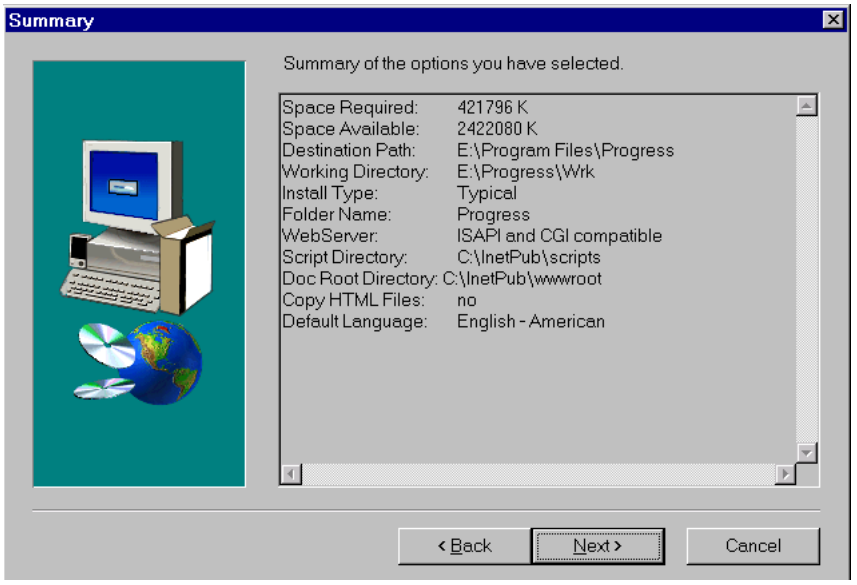
7 ♦ Choose Next. The International Settings dialog box appears:



The International Settings dialog box allows the installation program to tailor your Progress startup parameter (`startup.pf`) file to specify international conventions, such as Date format, Number format, Character Set, Collation, and Case preferences. For more information on international conventions, see the *Progress Internationalization Guide*.

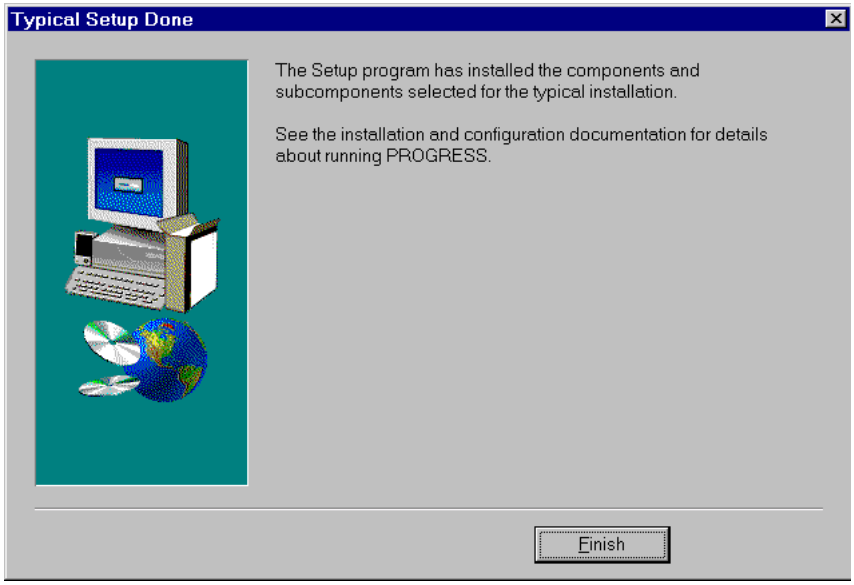
8 ♦ From each drop-down list, select the international Character Set, Collation, Case; Date Format; and Number Format to tailor your Progress `startup.pf` file.

9 ♦ Choose Next. The Summary dialog box appears, displaying a summary of the installation options you selected:



10 ♦ Choose Next. The setup program installs the WebSpeed files.

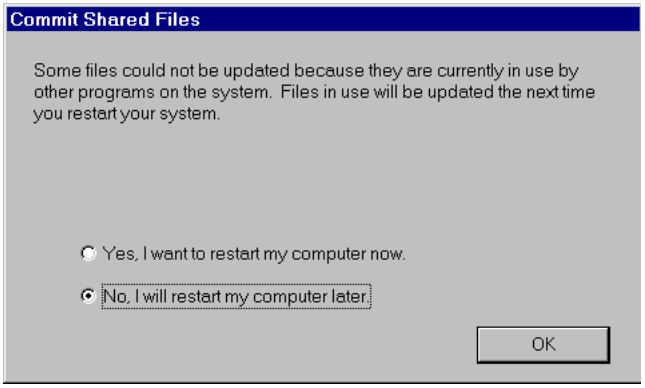
11 ♦ When the installation finishes, the Setup Done dialog box appears:



The Progress installation updates shared files that are commonly in use by programs on your system. If no shared files were in use during the installation, the Setup Done dialog box provides a toggle box that you use to launch the Progress Explorer and connect to an AdminServer. You can use the Progress Explorer tool to control and configure your WebSpeed environment. (If you do not want to configure Progress now, deactivate the toggle box.)

If shared files were in use during the installation, you do not have the option to start the Progress Explorer and the Commit Shared Files dialog box displays when you choose Finish. You cannot configure WebSpeed until the shared files are updated.

12 ♦ Select the appropriate radio button on the Commit Shared Files dialog box and choose OK:



After the shared files are updated, the installation automatically starts the AdminServer and the predefined NameServer.

On Windows NT, the AdminServer is configured 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 tools and databases.

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

For more information on the AdminServer and Progress Explorer, see Chapter 2, "Overview Of the WebSpeed Architecture."

You are now ready to verify the installation by configuring and running the WebSpeed Broker and running the test procedure. For information, see Chapter 7, "Configuring WebSpeed On Windows."

5.3 Performing a Custom Installation

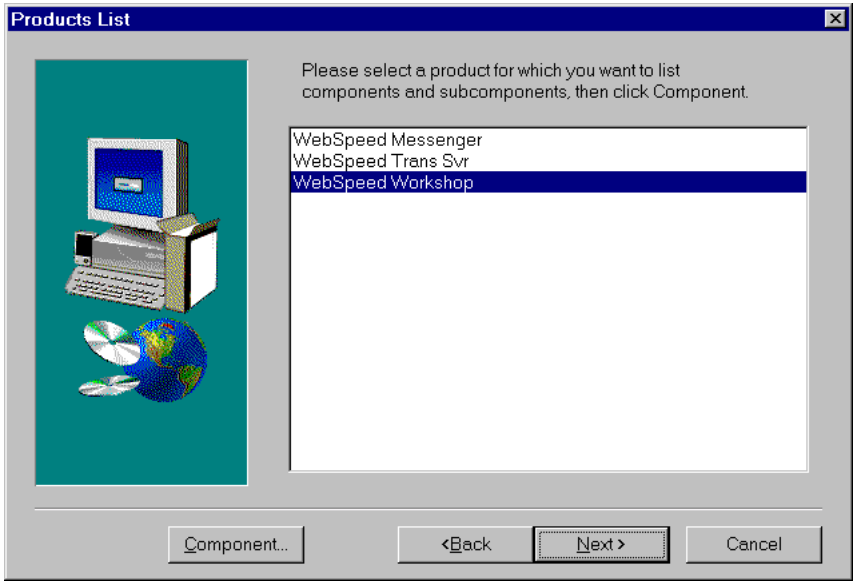
Perform a custom installation when you want to install individual WebSpeed components. For example, the WebSpeed Messenger must reside on the same machine as your Web server. If you installed WebSpeed Workshop or WebSpeed Transaction Server on another machine, you must run the setup program on your Web server machine to install the WebSpeed Messenger. See Chapter 3, "WebSpeed Configurations," for information on distributing WebSpeed components.

Installing individual components might also help you work around disk space limitations. You can then choose to install only the mandatory WebSpeed components. Plan your Custom Installation carefully before you start. If you later find that you need a component or subcomponent that you did not install, you can add it to the installation using the Add Component feature.

Follow these steps to perform a Custom Installation:

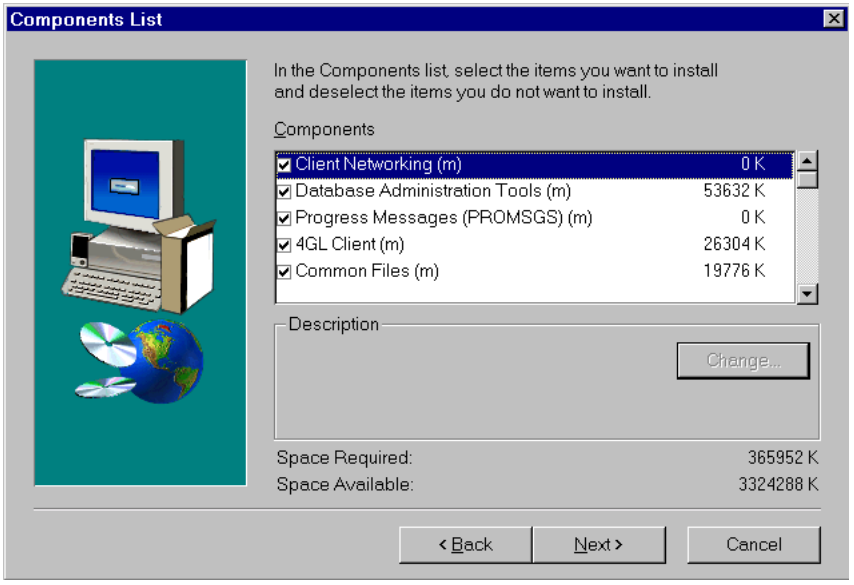
1 ♦ Complete the steps in the "Beginning the Installation" section.

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 individual components you want to be able to install or not install.

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

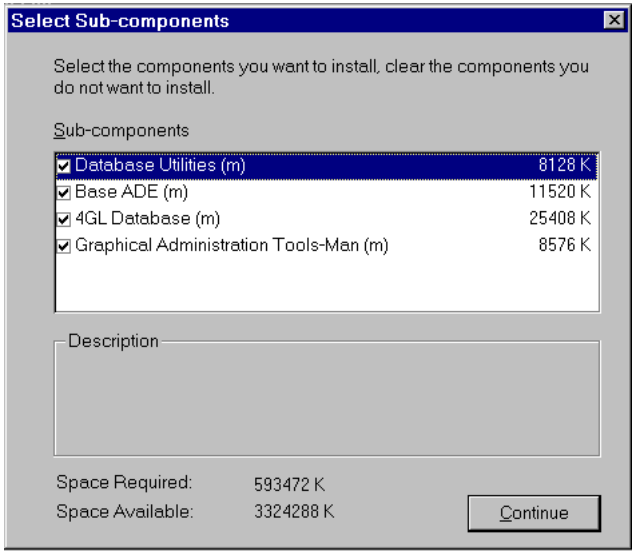


Some of the components are marked as mandatory. You must include these components in your WebSpeed installation, though not necessarily on the machine you are currently setting up. You should ignore the warning messages about installing mandatory components only if you are an experienced WebSpeed user.

5 ♦ Select the components for a custom installation.

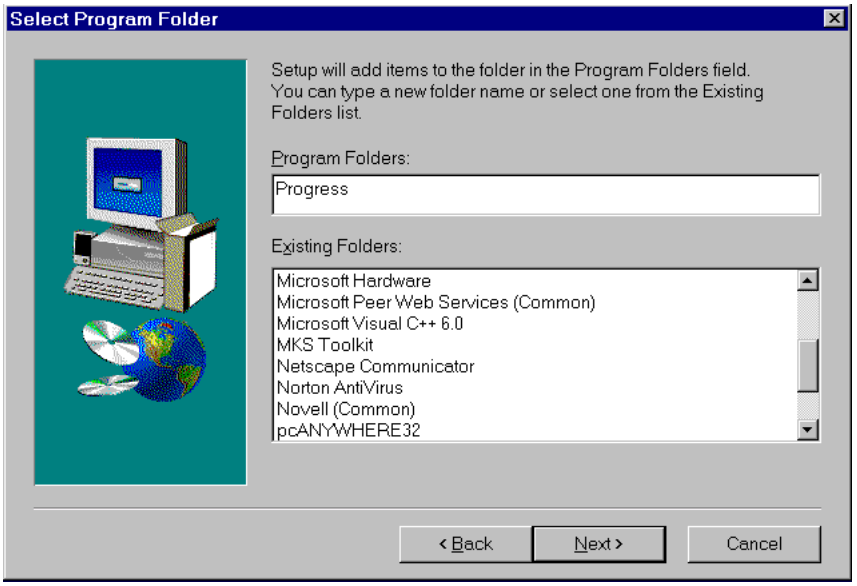
NOTE: If you previously installed the required version of the JavaSoft (InstallShield) JDK on your system and want Progress and WebSpeed to use the installed version, you should contact Progress Technical Support for assistance.

6 ♦ Choose Change. The Select Sub-components dialog box appears:



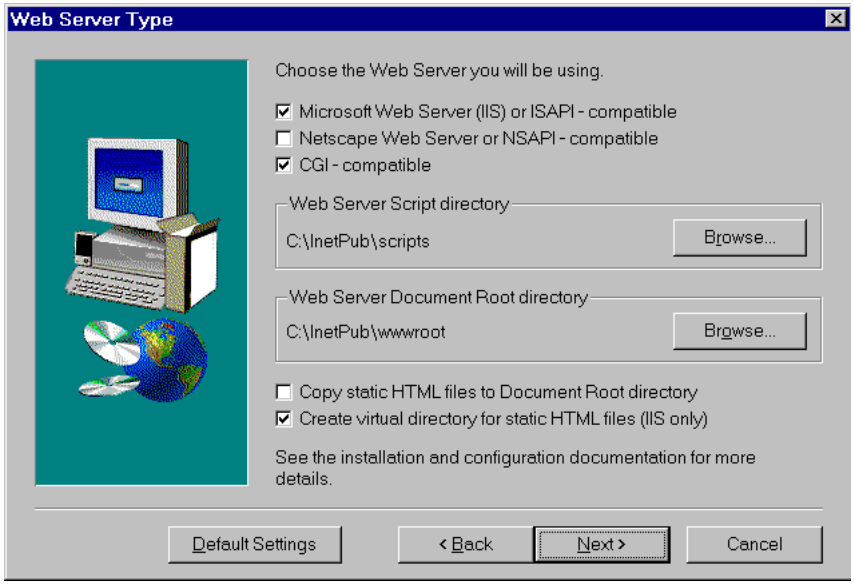
Some of the subcomponents are marked as mandatory. You must include these subcomponents in your WebSpeed installation, though not necessarily on the machine you are currently setting up. You should ignore the warning messages about installing mandatory subcomponents only if you are an experienced WebSpeed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

7 ♦ Select the subcomponents that you want to install and choose Next. The Select Program Folder dialog box appears:



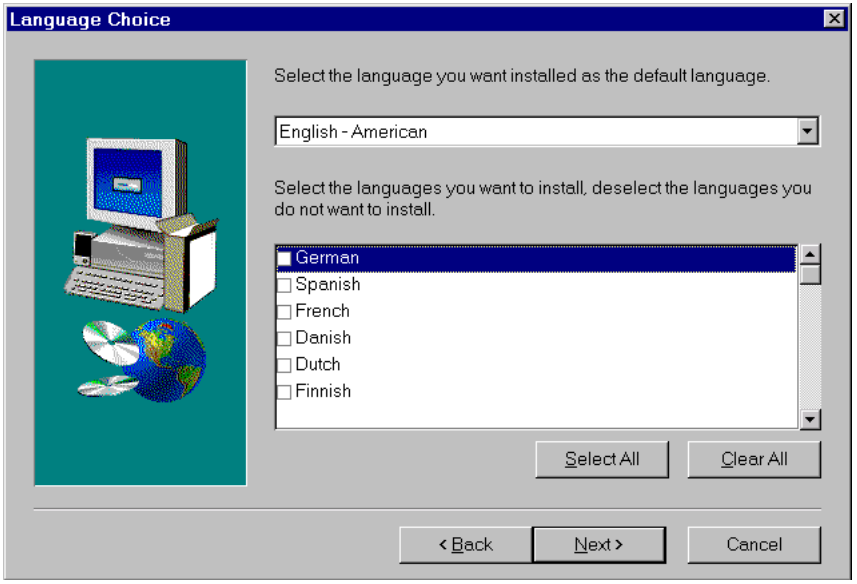
8 ♦ Select the program folder that will contain the WebSpeed files.

9 ♦ Choose Next. If you are installing a WebSpeed messenger component, the Web Server Type dialog box appears:



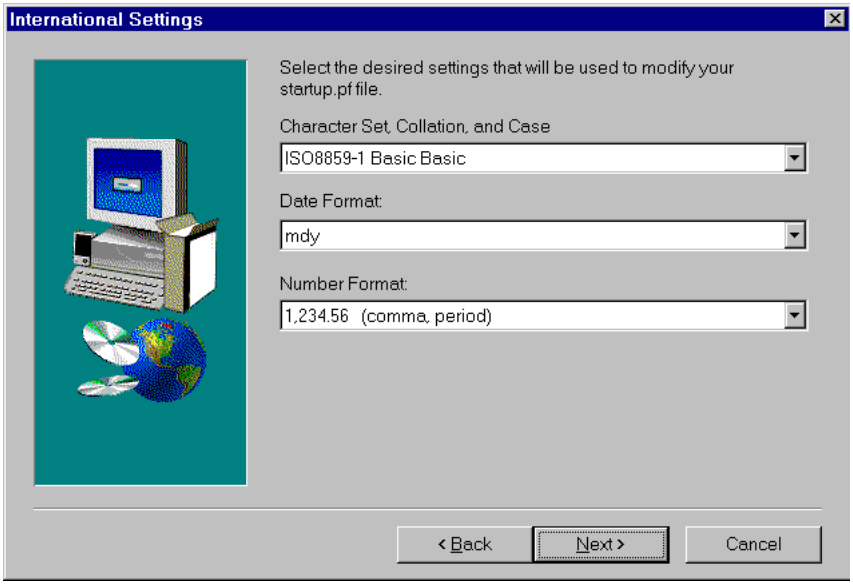
10 ♦ Select the Web Server for which you want to install Messengers. Choose Next.

The Language Choice dialog box appears:



11 ♦ Select the default language and any other languages you want to install. WebSpeed can use these languages for run-time messages. The initial value or default language is determined by the operating system on which you are installing WebSpeed.

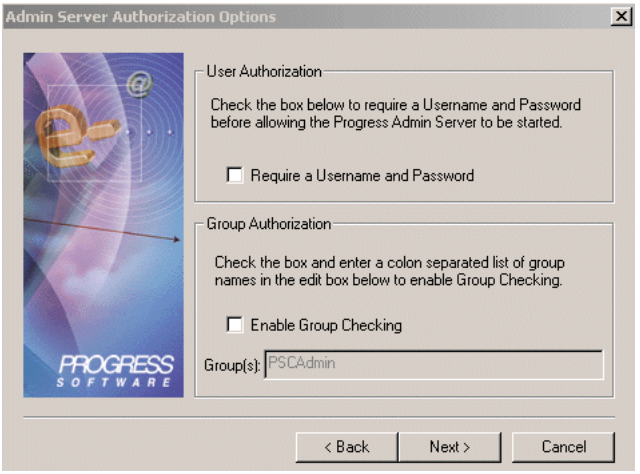
12 ♦ Choose Next. The International Settings dialog box appears:



The International Settings dialog box allows the installation program to tailor your Progress startup parameter (`startup.pf`) file to specify international conventions, such as Date format, Number format, Character Set, Collation, and Case preferences. 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 your Progress `startup.pf` file.

14 ♦ 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.

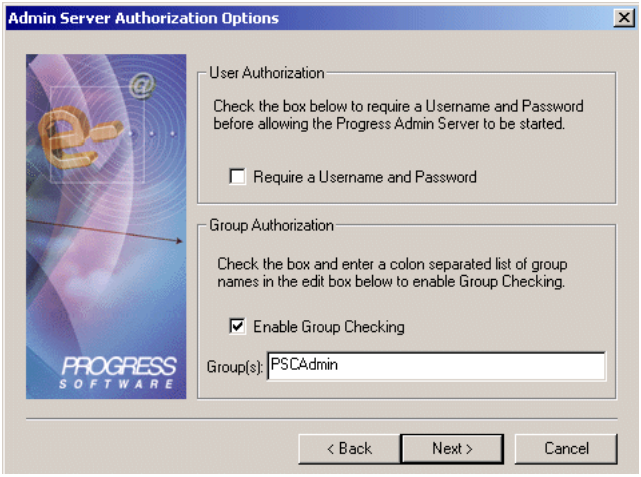
NOTE: The User Authorization and Group Authorization options are established independently of each other. You can elect to use one option, both options, or none. To bypass this option, select Next and proceed to Step 19 in this procedure.

15 ♦ 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 initiate 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.

16 ♦ 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 space or spaces you enter prior to your 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 5–1.

Table 5–1: Illegal Characters

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

17 ♦ 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 5–1 shows the Question dialog box that displays for the AdminGroup1 entry.

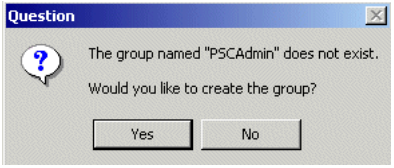


Figure 5–1: Question Dialog Box

18 ♦ 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 5–2 shows the Warning dialog box that appears if you select No.

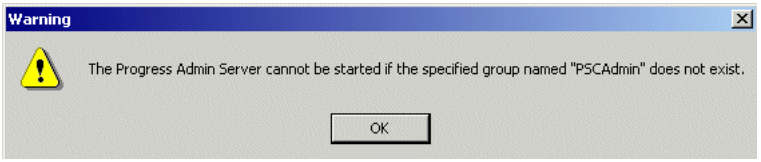
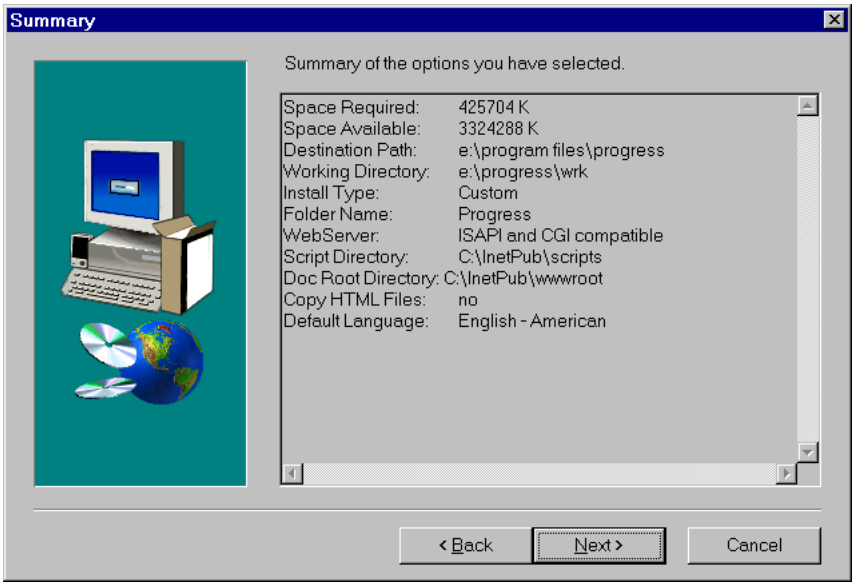


Figure 5–2: Warning Dialog Box

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:

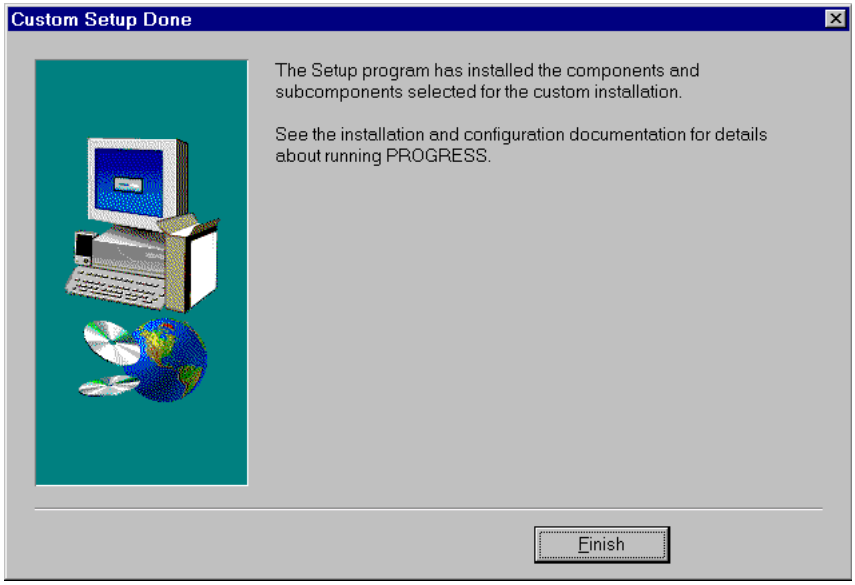
- Re-install WebSpeed using the installation program
- Add new groups when the Admin Server Authorization Options dialog box appears.
- Repeat steps 10 through 12 for each new group you enter.

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



20 ♦ Choose Next. The setup program installs the WebSpeed files.

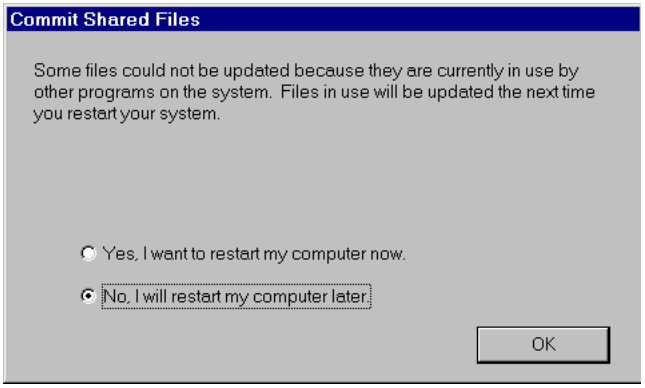
21 ♦ When the installation finishes, the Setup Done dialog box appears:



The installation updates shared files that are commonly in use by programs on your system. If no shared files were in use during the installation, the Setup Done dialog box provides a toggle box that you use to launch the Progress Explorer and connect to an AdminServer. You can use the Progress Explorer tool to control and configure your Progress environment. (If you do not want to configure WebSpeed now, deactivate the toggle box.)

If shared files were in use during the installation, you do not have the option to start the Progress Explorer and the Commit Shared Files dialog box displays when you choose Finish. You cannot configure Progress until the shared files are updated.

22 ♦ Select the appropriate radio button on the Commit Shared Files dialog box and choose OK:



After the shared files are updated, the installation automatically starts the AdminServer and the predefined NameServer.

On Windows NT, the AdminServer is configured 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 tools and databases.

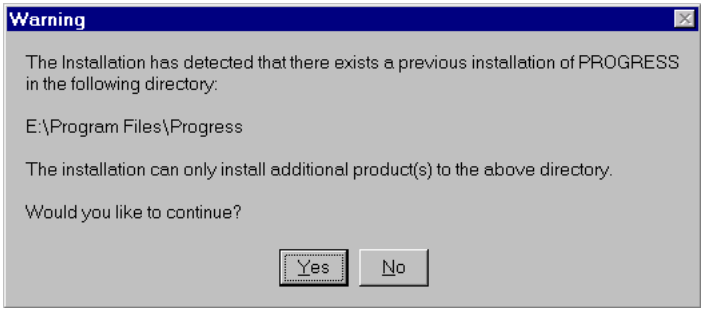
NOTE: Before you can start WebSpeed, the AdminServer must be running.

For more information on the AdminServer and Progress Explorer, see Chapter 2, "Overview Of the WebSpeed Architecture."

You are now ready to verify the installation by configuring and running the WebSpeed Broker and running the test procedure. For information, see Chapter 7, "Configuring WebSpeed On Windows."

5.4 Installing Additional Products

You can add other WebSpeed or Progress products to your current installation by following the steps outlined in the "Beginning the Installation" section of this chapter. The following dialog box appears:



Follow these steps to continue with the installation:

1 ♦ Click 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 for the products you want to add to your current installation.

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

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

NOTE: The install program deactivates (grays out) the Choose Destination And Working Path Directories dialog box modifier functionality and adds your Progress products to Version 9.1A directories automatically.

5.5 Network Installations

You can use the Shared Network Installation utility to provide multiple client access to a single copy of Progress or WebSpeed that is preinstalled on a network-accessible Windows 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.

For more information about network installations, see the *Progress Installation and Configuration Guide Version 9 for Windows*.

5.6 The InstallShield Silent (Batch Mode) Utility

The standard, interactive installation receives necessary information by prompting you and recording your input from dialog boxes. By contrast, a silent or batch mode installation does not prompt you for input. Instead, InstallShield Silent (ISS) reads input information from a file called a response file. You must create the response file before running a silent or batch installation.

5.6.1 Creating the Response File Automatically

The preferred method for creating a response file is to create it automatically.

In order to create a response file automatically, run the **setup.exe** with the **-r** option. Note the following syntax:

```
setup.exe -r [-f1C:\pathname\ResponseFile]
```

By default, the response file is created in your Windows directory and is named `setup.iss`. You can use the **-f1** option to specify an alternate directory and file name for the response file.

NOTE: Be sure that there is no space between **-f1** and `C:\pathname\ResponseFile`.

When you run **setup.exe** with the **-r** option, InstallShield runs the installation interactively and creates the response file, which contains all of your installation choices.

5.6.2 Creating the Response File Manually

You can create and edit a response file in any text editor by adding necessary sections in a predefined order.

Follow these steps to create a response file manually:

1 ♦ Create a new text file using any text editor.

2 ♦ Enter required sections in the following order, noting that section names are contained in square brackets:

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 there response file.

The following shows what a response file looks like:

Sample Response File *(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
```

Sample Response File *(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
```

5.6.3 Running the InstallShield Silent Utility

Follow these steps to run the InstallShield Silent utility:

1 ♦ Create a Response file.

For information on creating a response file, see "Creating the Response File Automatically," or "Creating the Response File Manually."

2 ♦ Run **setup.exe** using the **-s** option.

Notice the following syntax:

```
setup.exe -s [-f1C:\pathname\ResponseFile] [-f2C:\pathname\SetupLog]
```

By default, InstallShield looks for a response file named `setup.iss` in your Windows directory. During the installation, a log file named `setup.log` is created in your Windows directory. Use the **-f1** option to specify an alternate directory and file name for the response file. Use the **-f2** option to specify an alternate directory and file name for the log file.

NOTE: Be sure that there is no space between **-f1** or **-f2** and the pathnames that follow.

5.7 Uninstalling WebSpeed

When you arbitrarily delete files from the Progress directory tree, you only partially remove the WebSpeed installation. By contrast, the Uninstall or the Remove Program utility removes all WebSpeed files as well as configuration information from the registry, which prevents conflict with subsequent WebSpeed installations.

5.7.1 Before You Begin

Before you begin the uninstallation process, follow these steps:

1 ♦ Make sure that your Microsoft IIS or Netscape Web server is shut down.

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

3 ♦ From the Windows desktop, select Start→ Settings→ Control Panel→ Services.

4 ♦ Highlight the Progress AdminService, and choose Startup. The Service dialog box appears.

5 ♦ Change the startup from Automatic to Manual, and choose OK.

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

5.7.2 Using the Uninstall Or Add/Remove Program Utility

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

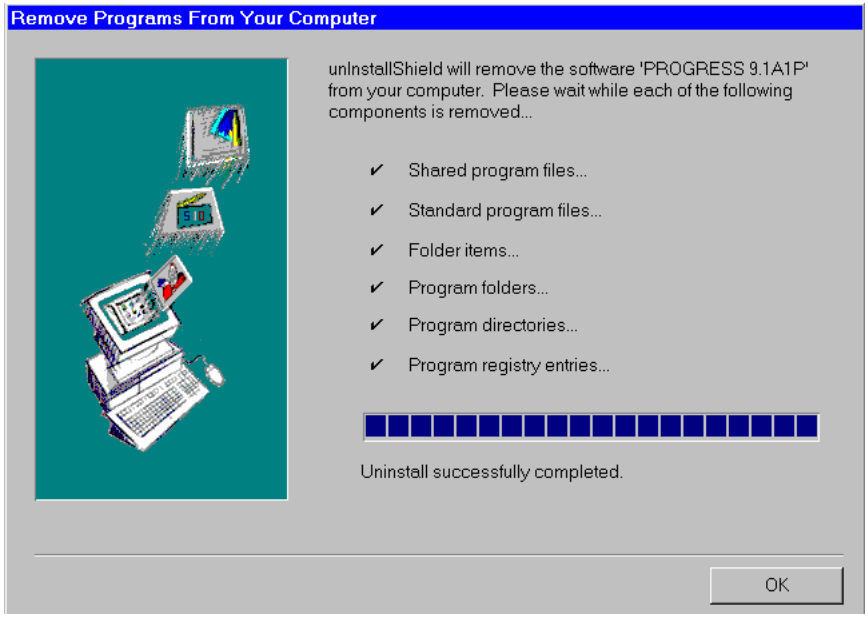
Follow these steps to run the Uninstall utility:

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

2 ♦ Choose the Uninstall icon in the Progress Program Group. The Confirm File Deletion dialog box appears:



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



The Uninstall utility deletes the components that you installed.

4 ♦ Choose OK when the uninstall process is complete.

6

Installing WebSpeed On UNIX

This chapter contains instructions for installing WebSpeed on a UNIX platform. The chapter contains information on the following topics:

- Beginning the Installation
- Performing a Typical Or a Complete Installation
- Performing a Custom Installation
- Running the Installation Utility In Batch Mode

6.1 Beginning the Installation

How you install WebSpeed depends on what products, components, and subcomponents you choose to install. You can tailor your installation so that only those products and components essential to your business operation are installed.

As you proceed through the installation, you will be asked to choose one of the following installation options:

- **Typical**—Installs the most commonly used WebSpeed components and subcomponents. This is the default installation.
- **Complete**—Installs all WebSpeed components and subcomponents.
- **Custom**—Installs only the WebSpeed components and subcomponents that you choose.

Read the system requirements information in Chapter 1, "Installation Requirements," before performing the installation.

Follow these steps to begin installing your WebSpeed products:

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

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

3 ♦ Log in as root. If you do not know the root password for your machine, check with your system administrator.

4 ♦ Insert the installation CD into the CD-ROM drive.

NOTE: You cannot install WebSpeed on an NFS mounted device.

5 ♦ Enter your platform-specific mount command (where *device-name* is the device you are using for the installation and *mount-point* is the mount-point directory). Table 6–1 lists the mount commands for each supported platform.

Table 6–1: Mount Commands

Operating System	Mount Command
HP-UX 11 (32-bit)	`mount -F cdfs -r -o cdcase` *device-name mount-point* For example: `mount -F cdfs -r -o cdcase /dev/dsk/c0t2d0 /cdrom`
IBM AIX 4.3.2 (32-bit)	`mount -v cdrfs -r` *device-name mount-point* For example: `mount -v cdrfs -r /dev/cd0 /cdrom`
SCO UnixWare 7.0	`mount -F cdfs -r -o nmconv=lm,fperm=555,dperm=555` *device-name mount-point* For example: `mount -F cdfs -r -o nmconv=lm,fperm=555,dperm=555 /dev/cdrom/c0b0t210 /cdrom`
Sun Solaris SPARC 2.6, 2.7 (32-bit)	`mount -F hsfs -o ro,nrr -r` *device-name mount-point* For example: `mount -F hsfs -o ro,nrr -r /dev/dsk/c0t4d0s0 /cdrom`
Sun Solaris Intel 2.6 and 2.7	`mount -F hsfs -r -o ro,nrr` *device-name mount-point* For example: `mount -F hsfs -r -o ro,nrr /dev/dsk/c0t4d0s0 /cdrom`
Compaq Tru64 UNIX 4.0D	`mount -t cdfs -o noversion` *device-name mount-point* For example: `mount -t cdfs -o noversion /dev/rz3c /cdrom`

6 ♦ Enter the following install command:

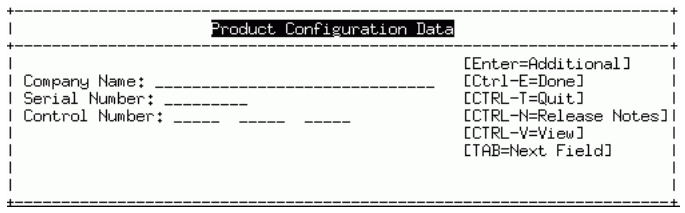
```
mount-point/proinst
```

NOTE: You cannot run `proinst` if you are in the mount-point directory.

The Welcome dialog box appears:



7 ♦ Press **RETURN**. The Product Configuration Data dialog box appears:



The information you provide depends on whether you are installing all WebSpeed components on the same machine or installing a subset of WebSpeed components and distributing WebSpeed across networked machines.

8 ♦ Type in your company name (any character string is acceptable), the serial number, and the control number from your *License Addendum* for the first product you want to install. Use the **TAB** key to move to the next line.

This information is not case sensitive. The installation process automatically converts all letters to uppercase.

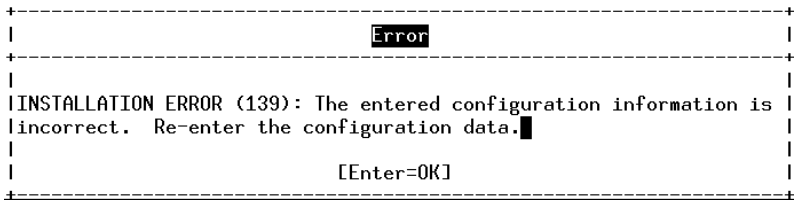
Table 6–2 describes the options available in the Product Configuration Data dialog box.

Table 6–2: Product Configuration Data Dialog Box Options

Option	Function
ENTER = Additional	Confirms that the product information entered is correct and clears the input fields for additional information for another WebSpeed product you want to install. If you enter incorrect information, an error message appears indicating that the information is invalid. Press **RETURN**, correct the product information, and continue with the installation.
CTRL-E = Done	Confirms that the product information entered is correct, and continues to the next step in the installation. The actual key you use to indicate Done varies by terminal type. Refer to the Product Configuration Data dialog box for the appropriate key combination for your terminal. If you enter incorrect information, an error message appears indicating that the information is invalid. Press **RETURN**, correct the product information, and continue with the installation.
CTRL-T = Quit	Quits the Installation Utility. The Quit Confirmation dialog box appears. Choose **Y** to quit or **N** to continue the installation. (When you quit the installation process, the installation utility and all the installation files you copied into your working directory are deleted.)
CTRL-N = Release Notes	Displays the online *Release Notes*. Opens the `vi emacs` editor, which displays the WebSpeed Web Site URL where you can access the *Release Notes* for the installation. To close the editor, press **:q!** or **ESCAPE ZZ**.
CTRL-V = View	Displays the selected WebSpeed products for which you entered control numbers.
TAB = Next Field	Moves to the next field in the dialog box.

9 ♦ To select an option, highlight the option using the **SPACEBAR**, **TAB**, cursor keys, or the accelerator keys that are highlighted in each selection. Press **RETURN** to execute the selected option.

If you enter incorrect product configuration information, the following error message appears indicating that the data is invalid:



Press **RETURN** to correct the product information and continue with the installation. The Product Configuration Data dialog box reappears, allowing you to enter the configuration data for the next product you want to install.

10 ♦ To view the WebSpeed products for which you entered control numbers, press **CTRL-V** (see Table 6–2):



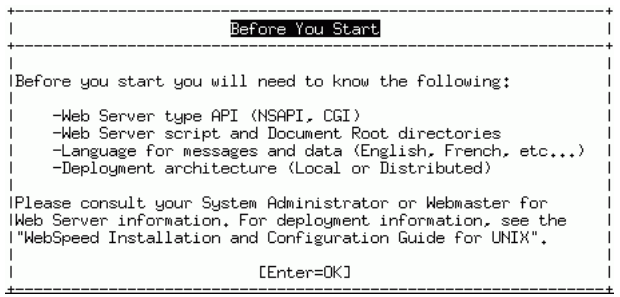
After viewing the product control numbers that you entered, press **RETURN** and the Product Configuration Data dialog box reappears, allowing you to enter the configuration data for the next product you want to install.

11 ♦ When you finish entering the control numbers for the products you want to install, press **CTRL-E** (see Table 6–2). The Done Configuration Data Confirmation dialog box appears:



12 ♦ Type **Y** to continue with the installation.

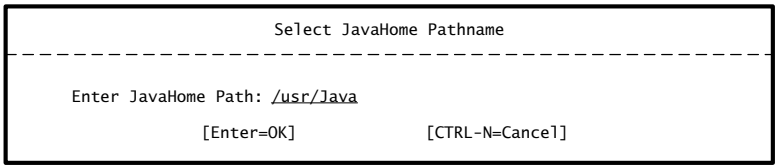
13 ♦ If you are installing a WebSpeed product that includes a messenger component, the Before You Start dialog box appears:



To continue with the installation, you must be able to provide the following information:

- **Web Server type**—You must know whether the Web server that you are using with WebSpeed is NSAPI- or CGI-compatible. Choose which WebSpeed Messenger to install based on the kind of Web Server you have installed.
- **Web Server script directory**—You must know the absolute pathname of the directory that your Web server uses to store scripts. The installation utility installs the WebSpeed CGI Messenger script into that directory. It installs the Messenger executables in *install-path*/bin.
- **Web Server Document Root directory**—You must know the absolute pathname of the directory that your Web server uses to locate static HTML files.
- **Language**—You must decide what language you want WebSpeed to use when displaying messages and handling data.
- **Deployment Architecture**—You must decide whether you want to install all WebSpeed components on one machine or distribute them across two or more machines.

14 ♦ Press **RETURN** to continue. For those UNIX platforms on which the JRE/JDK is not automatically included during installation, the following dialog box, or one that is similar, appears:



The JavaHome directory is the Java installation directory.

NOTE: On these platforms, the installation process searches for the correct version of the JDK residing on the target system before continuing the installation.

15 ♦ Enter a different version/location than the one specified or press **RETURN** to continue.

The Type, Device and Destination dialog box appears:

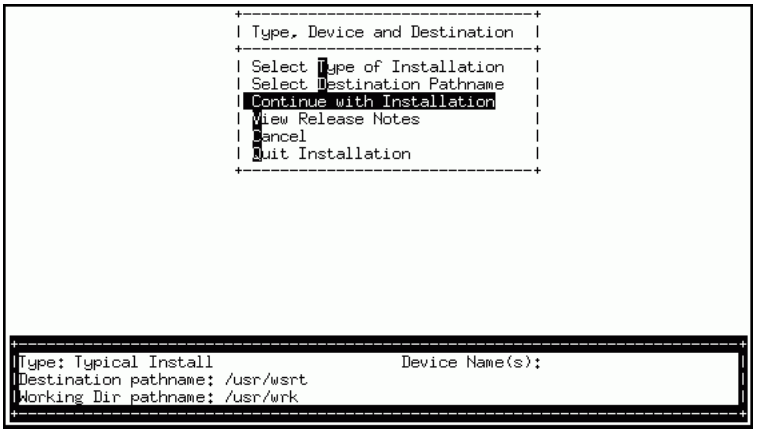


Table 6–3 describes the menu options in the Type, Device and Destination dialog box. To select an option, highlight the option using the **SPACEBAR**, **TAB**, cursor keys, or the accelerator keys that are highlighted in each selection. Press **ENTER** to execute the selected option.

Table 6–3: Type, Device and Destination Dialog Box Options

Option	Function
Select Type of Installation	Allows you to select the type of installation you want to perform: Typical, Complete, or Custom.
Select Destination Pathname	Allows you to change the default destination pathname (`/usr/dlc`).[1]
Continue with Installation	Confirms your selections (or accepts the defaults listed in the status box) and continues with the installation.
View Release Notes	Views the latest release notes using the `vi` or `emacs` editor. When you choose this option, the Installation Utility launches the editor. To exit `vi` after reading the *Release Notes*, press **:q!**. To exit `emacs`, press **CTRL-X** and then **CTRL-C**
Cancel	Cancels the previous selection and returns you to the Product Configuration Data dialog box.
Quit Installation	Quits the Installation Utility. The Quit Confirmation dialog box appears. Choose **Y** to quit or **N** to continue the installation. (When you quit the installation process, the installation utility and all the installation files you copied into your working directory are deleted.)

[1] If you have Progress or WebSpeed products installed, do not enter the pathname where these products reside.

CAUTION: Do not make your working directory a subdirectory under the installation path. If you uninstall WebSpeed, the working directory is deleted along with the destination directory.

NOTE: Progress Software Corporation recommends that you read both the online and hard-copy *Release Notes* before continuing with the installation. The *Release Notes* replace or supplement the installation instructions in this booklet. The online *Release Notes* appear in the `vi` or `emacs` editor.

6.2 Performing a Typical Or a Complete Installation

Follow these steps to choose the Typical or the Complete installation option:

1 ♦ Choose Select Type of Installation. The following dialog box appears:

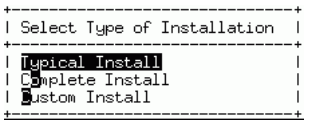


Table 6–4 describes the installation options.

Table 6–4: Installation Options

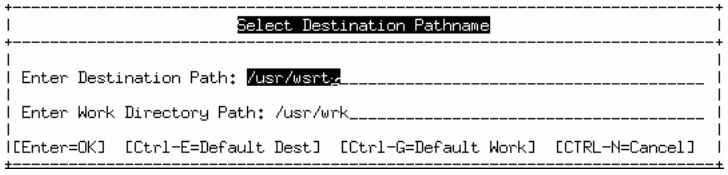
Option	Description
Typical	Choose this installation method when you want to install the most common (typical) WebSpeed components and subcomponents. This default installation option is recommended for most users.
Complete	Choose this installation method when you want to install all WebSpeed components and subcomponents.
Custom	Choose this installation method when you want to distribute WebSpeed components on different machines. Installing individual components might also help you work around disk space limitations by allowing you to install only mandatory components. This method is recommended for advanced users. If you want to perform a Custom installation, continue with the "Performing a Custom Installation" section.

2 ♦ Select either a Typical or a Complete installation as the type of installation you want to perform, and press **RETURN**. The Type, Device and Destination dialog box reappears.

6.2.1 Selecting the Destination Pathname

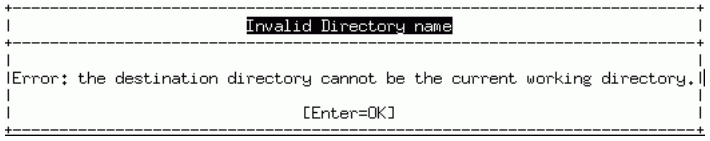
Follow these steps to continue with the installation:

1 ♦ Choose Select Destination Pathname. The following dialog box appears:



2 ♦ Type your Progress destination path or accept the default directory in which to install Progress (`/usr/dlc`). The actual key you use to accept the default varies by terminal type. Refer to the Select Destination Pathname dialog box for the appropriate key combination for your terminal.

You cannot use the current working directory as the destination directory. If you type the same path for Destination Path and Work Directory Path, the following error message appears:



Once you choose an option on the Destination Pathname dialog box, press **RETURN**. The Select Destination Pathname dialog box reappears.

Choose a different pathname.

3 ♦ If the directory does not have enough space to install the products that you chose, the following dialog box appears:



When you press **RETURN**, the Select Destination Pathname dialog box reappears.

4 ♦ If the directory you chose for the installation already exists, the following dialog box appears:

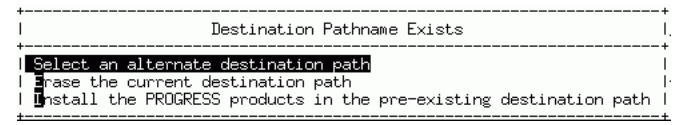


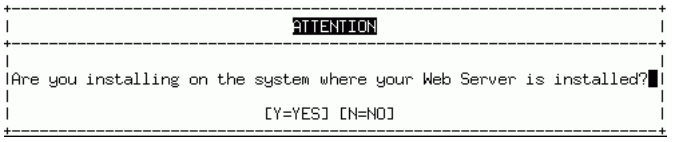
Table 6–5 lists the options in the Destination Pathname Exists dialog box.

Table 6–5: Destination Pathname Exists Dialog Box Options

Option	Action
Select an alternate destination path	Allows you to select another destination path. This returns you to the Select Destination Pathname dialog box.
Erase the current destination path	Deletes the contents and subdirectories of the named directory and reinstalls in the same directory.
Install the WebSpeed products in the pre-existing destination path	Installs WebSpeed in the named directory. The installation utility overwrites only filenames matching those it will install and updates, but does not overwrite, the `progress.cfg` file. **CAUTION:** When you perform a product update, the existing `ubroker.properties` file is saved with `-save` appended to the filename; and a new `ubroker.properties` file is created. This process repeats each time you perform a product update. Because only the current version of the file is saved when you perform an update, you might want to save a copy of your original `ubroker.properties` file in the *`install-path`*`/properties` directory before you perform an update.

5 ♦ When you have a directory to which you can install the products, press **RETURN** from the Select Destination Pathname dialog box. The Type, Device and Destination dialog box reappears.

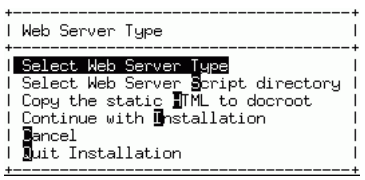
6 ♦ At the Type, Device and Destination dialog box, choose Continue with Installation. A message appears asking you if you are installing on the system where your Web Server is installed:



7 ♦ Type **Y** or **N**.

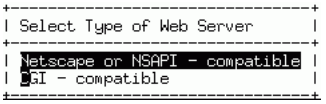
If you type **N**, proceed to Step 11.

If you type **Y**, the Web Server Type dialog box appears:



8 ♦ By default, the installation utility assumes that you are using a Netscape Web server.

- If you are using a Netscape Web server, proceed to Step 11.
- If you want install the Messenger executable and script for a CGI-compatible Web server, choose Select Web Server Type:



Choose CGI-compatible and press **RETURN**. The Web Server Type dialog box reappears.

9 ♦ From the Web Server Type dialog box, choose Select Web Server Script directory. This step is required only if you are installing a CGI Messenger. The script for the CGI Messenger must reside in your Web server's scripts directory. The Select Web Server Script directory dialog box appears:



10 ♦ Enter the pathname of your Web server's scripts directory, then press **RETURN**.

11 ♦ From the Web Server Type dialog box, choose Copy the static HTML to docroot if you want the installation utility to copy the static files required by Workshop into your Web server's document root directory. The files will be copied from *install-path*/webspeed to the directory you specify in the following dialog box:



12 ♦ Enter the pathname of your Web server's document root directory and press **RETURN**. The Web Server Type dialog box reappears.

13 ♦ Choose Continue with Installation. The Language Selection dialog box appears:



14 ♦ Select the language you want to install as the default by using your arrow keys to highlight the language, then press **RETURN**. (You might need to use the down arrow key to reach the end of the list.) An asterisk (*) appears next to the selected language.

Press `M`, the Make Default quick key. A (d) appears next to the default language.

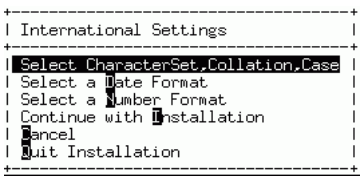
If you do not select a default language, the following error message appears:



To select another language to install, highlight the language and press **RETURN**.

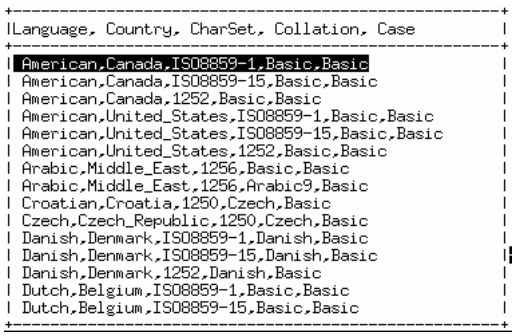
15 ♦ When you finish selecting languages, choose Continue with Installation.

The International Settings dialog box appears:



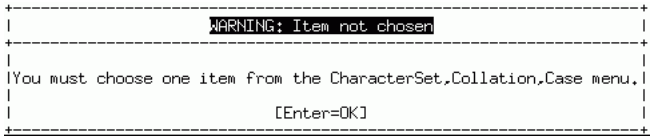
The International Settings dialog box allows you to tailor your Progress startup (`startup.pf`) file to accommodate international conventions such as Date (-d) format, Number (-E) format, Character Set, Collation, and Case preferences. For more information on international conventions, see the *Progress Internationalization Guide*.

16 ♦ Choose the Character Set, Collation, Case option. The Language, Country, CharSet, Collation, Case dialog box appears:



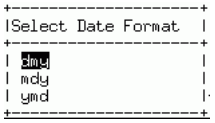
NOTE: You might need to use the down arrow key to reach the end of the list.

If you do not choose an item from this menu, the following error message appears when you continue the installation:

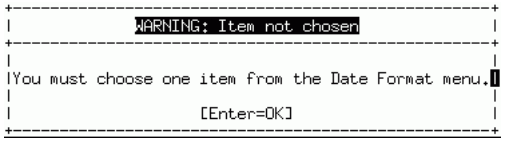


Press **RETURN** and complete the required setting.

17 ♦ Choose the Date Format option. The Select Date Format dialog box appears:

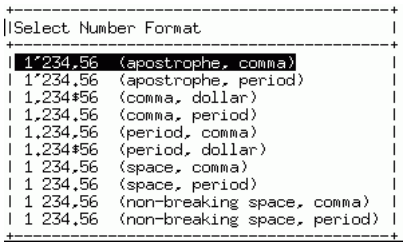


If you do not choose an item from this menu, the following error message appears when you continue the installation:

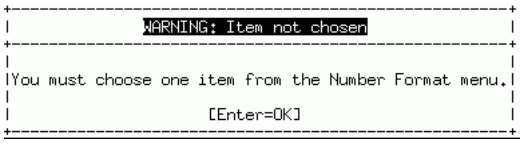


Press **RETURN**, and complete the required setting.

18 ♦ Choose the Number Format option. The Select Number Format dialog box appears:

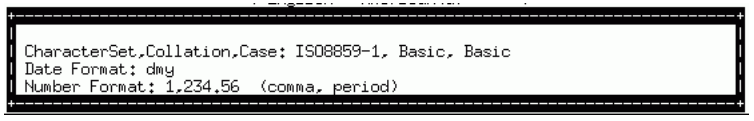


If you do not choose an item from this menu, the following error message appears when you continue the installation:



Press **RETURN**, and complete the required setting.

19 ♦ As you select options in the International Settings dialog box, the Status bar at the bottom of the display shows your choices. Choose the Continue with Installation option when the Status bar shows a value for each of the settings:



20 ♦ Once you provide all required international settings, the installation performs a check of available disk space. If you have insufficient disk space, a message appears:



21 ♦ Press **RETURN**, and change the path. Provided your new destination choice has enough disk space, the Installation Summary dialog box appears, displaying a summary of the installation options you selected:

```
+-------------------------------------------------------------+
|                    Complete Installation                    |
+-------------------------------------------------------------+
|                                                             |
|The following products will be installed:                    |
|'WebSpeed Trans Svr', 'WebSpeed Messenger'                   |
|                                                             |
|Disk Space Required for Products: 156,590,080 bytes          |
|Disk Space Required for Installation: 157,499,392 bytes      |
|Disk Space Remaining After Installation: 620,286,976 bytes   |
|                                                             |
|Selected Destination Path: /usr1/pmccorma/wsrt               |
|                                                             |
|Do you want to install the above listed product(s)?          |
|                                                             |
|                       [Y=YES] [N=NO]                        |
+-------------------------------------------------------------+
```

22 ♦ Choose **Y** to continue. The Copy Scripts? dialog box appears:

```
+-----------------------------+
|        Copy Scripts?        |
+-----------------------------+
|                             |
|Copy the scripts to /usr/bin?|
|                             |
|       [Y=YES] [N=NO]        |
+-----------------------------+
```

WebSpeed consists of several scripts and program modules. When you install a product, the scripts are placed in the installation directory you specify.

23 ♦ To allow all users on your system to run the product, you should answer yes when prompted to copy the scripts to `/usr/bin`.

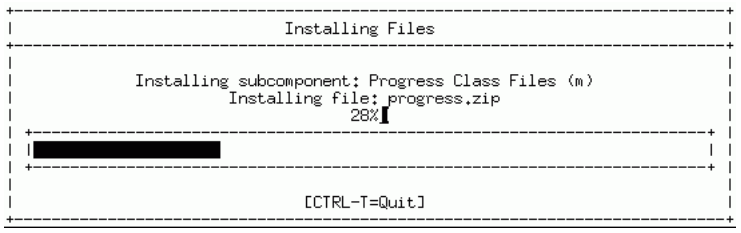
Choose **Y** to instruct the installation utility to place WebSpeed scripts in `/usr/bin` **and** the destination path you specified earlier in the installation.

Choose **N** to instruct the installation utility to place WebSpeed scripts **only** in the destination path you specified earlier in the installation.

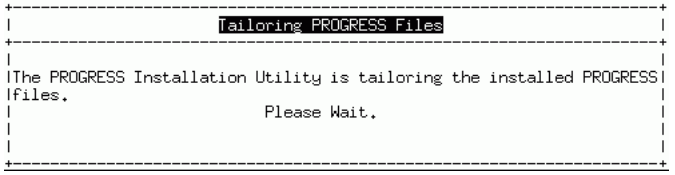
CAUTION: If you have other WebSpeed or Progress products already installed on this machine, answer **No** when prompted to copy the scripts to `/usr/bin`. If you answer **Yes**, the installation utility overwrites any existing Progress scripts in `/usr/bin`.

NOTE: If you are maintaining two versions of WebSpeed or Progress on the same machine, answer **Yes** to this question only if most users will be using the version you are now installing.

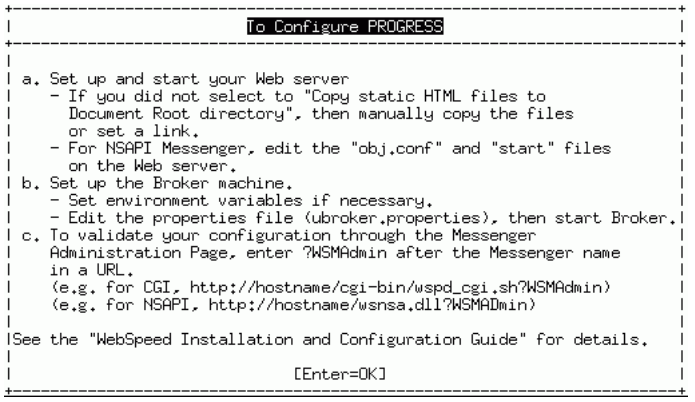
While the utility is decompressing the files, the following dialog box appears:



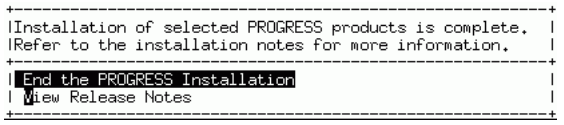
While the utility is tailoring the files, the following dialog box appears:



If you are installing a WebSpeed product such as the Messenger component, the To Configure Progress dialog box appears:



24 ♦ Press **RETURN**. When the installation is complete, the following dialog box appears:



25 ♦ Choose End the Progress Installation to exit. Progress returns you to the UNIX system prompt.

26 ♦ You can now unmount the CD by entering the following command:

```
umount device-name mount-point
```

You are now ready to verify the installation by configuring and running the WebSpeed Broker and running the test procedure. For information, see Chapter 8, "Configuring WebSpeed On UNIX."

6.3 Performing a Custom Installation

Perform a custom installation when you want to install individual WebSpeed components. For example, the WebSpeed Messenger must reside on the same machine as your Web server. If you installed WebSpeed Workshop or WebSpeed Transaction Server on another machine, you must run the installation utility on your Web server machine to install the WebSpeed Messenger. See Chapter 6, "Installing WebSpeed On UNIX," for information on distributing WebSpeed components.

Installing individual components might also help you work around disk space limitations. You can then choose to install only the mandatory WebSpeed components. Plan your custom installation carefully before you start. If you later find that you need a component or subcomponent that you did not install, you can add it to the installation using the Add Component feature.

To perform a Custom Installation, follow these steps:

1 ♦ Follow the steps in the "Beginning the Installation" section. The Type, Device and Destination dialog box appears.

The message area at the bottom of the dialog box shows the default settings for the installation. The default installation type is Typical, and the destination pathname is `/usr/dlc`. (The Typical Install installs all WebSpeed files and executables without interfering with any other WebSpeed or Progress products you might have installed previously.)

2 ♦ On the Type, Device and Destination dialog box, select Custom Installation. Press **RETURN**. The Type, Device and Destination dialog box reappears.

3 ♦ Select Continue with Installation. The Custom – Select Products dialog box appears:



Each product you are installing is preceded by an asterisk (*).

Table 6–6 describes the status bar at the bottom of the Custom – Select Products dialog box.

Table 6–6: Custom – Select Products Dialog Box Status Information

Status Data	Description
Disk Space Required for Products	Displays the amount of disk space required to run the products you want to install.
Disk Space Required for Installation	Displays the amount of disk space required to install the products you want to install. This includes the space required to run the products plus additional space used and then released by the installation utility.
Disk Space Remaining After Installation	Displays the amount of disk space available after the actual installation. This includes the available disk space prior to installation, minus the disk space required for the products.
Destination pathname	Displays the destination pathname. If your destination pathname is truncated, select Change Destination Pathname to see the full pathname and verify its accuracy.

NOTE: If the Disk Space Required for Installation is greater than the disk space available on the destination drive, you have two options:

- Deselect WebSpeed subcomponents to be installed until you have enough available space.

- Select Change Destination Pathname to specify a different destination drive that has more available disk space. Press **RETURN**. The Destination Pathname Exists dialog box appears:

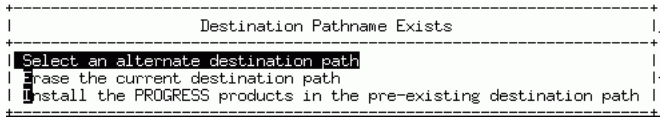


Table 6–7 lists the options in the Destination Pathname Exists dialog box.

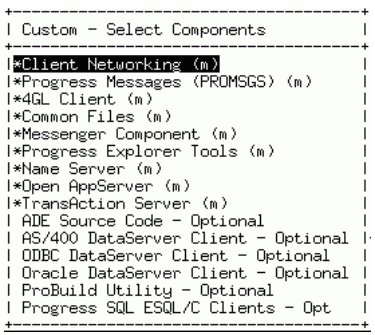
Table 6–7: Destination Pathname Exists Dialog Box Options

Option	Action
Select an alternate destination path	Allows you to select another destination path. This returns you to the Select Destination Pathname dialog box.
Erase the current destination path	Deletes the contents and subdirectories of the named directory and reinstalls in the same directory.
Install the WebSpeed products in the pre-existing destination path	Installs WebSpeed in the named directory. The installation utility overwrites only filenames matching those it will install. **CAUTION:** When you perform a product update, the existing `ubroker.properties` file is saved with `-save` appended to the filename; and a new `ubroker.properties` file is created. This process repeats each time you perform a product update. Because only the current version of the file is saved when you perform an update, you may want to save a copy of your original `ubroker.properties` file in the *`install-path`*`/properties` directory before you perform an update.

4 ♦ Select one of the options and continue as instructed. When you finish, the Custom – Select Products dialog box reappears.

5 ♦ Press **RETURN** to select or deselect a product. An asterisk (*) indicates that a product is selected.

6 ♦ Press **RETURN** (again) to customize the product. The Custom – Select Components dialog box appears:

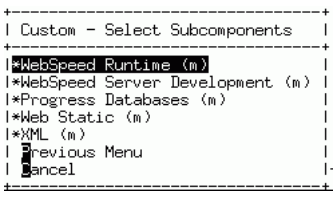


This dialog box lists the components of the selected product that you can install.

An (m) after a component indicates that the component (or subcomponent) is mandatory. In a distributed installation of WebSpeed, make sure that you install all mandatory requirements. It is not necessary to install all components on a single machine. The only component that must reside on each WebSpeed machine is the PROMSGS component. You can ignore the warning messages about installing mandatory components.

7 ♦ Select or deselect a component to install by highlighting the component and pressing **RETURN**.

8 ♦ Press **RETURN** (again) to view the subcomponents for the selected component. The Custom – Select Subcomponents dialog box appears:

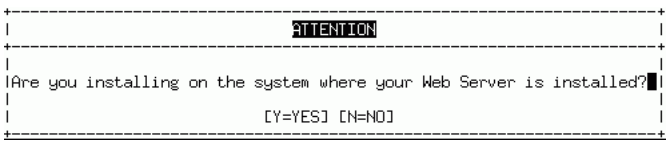


9 ♦ Select or deselect the WebSpeed subcomponents you want to custom install for the components you selected in Step 7.

10 ♦ Choose Previous Menu to return to the Custom – Select Components dialog box.

11 ♦ Choose Previous Menu to return to the Custom – Select Products dialog box.

12 ♦ At the Custom – Select Products dialog box, choose Install Selected Products. The following dialog box appears:



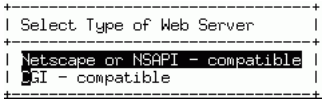
13 ♦ Type **Y** or **N**.

If you type **N**, proceed to Step 17.

If you type **Y**, the Web Server Type dialog box appears:



14 ♦ By default, the installation utility assumes that you are using a Netscape Web server. Choose Select Web Server Type if you want install the Messenger executable and script for a CGI-compatible Web server:



15 ♦ From the Web Server Type dialog box, choose Select Web Server Script Directory. This step is required only if you are installing a CGI Messenger. The script for the CGI Messenger must reside in your Web server's scripts directory. The Select Web Server Script Directory dialog box appears:



16 ♦ Enter the pathname of your Web server's scripts directory.

17 ♦ From the Web Server Type dialog box, choose Copy the static HTML to docroot if you want the installation utility to copy the static files required by the WebSpeed Workshop product into your Web server's document root directory. The files will be copied from *install-path*/webspeed to the directory you specify in the following dialog box:



18 ♦ Enter the pathname of your Web server's document root directory.

19 ♦ Choose Continue the Installation. The Language Selection dialog box appears:



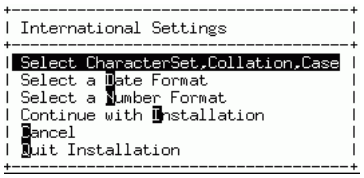
20 ♦ Select the languages for which you would like error messages installed by highlighting the language and pressing **RETURN**. An asterisk (*) appears next to the selected language. To select additional languages to install, highlight the languages and press **RETURN**.

21 ♦ You must select a default language. For you to select a default language, the language must be the last one you selected and marked with an asterisk. Then, highlight Make Default and press **RETURN**. A (d) appears next the default language.

If you do not select a default language, the following error message appears:

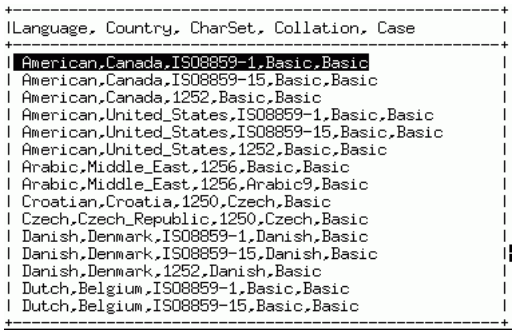


22 ♦ Choose Continue with Installation. The International Settings dialog box appears:



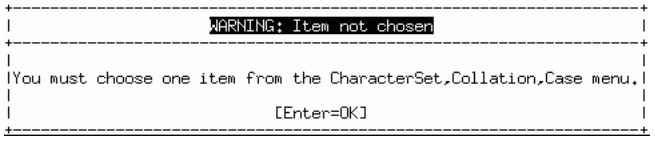
The International Settings dialog box allows you to tailor your Progress startup (`startup.pf`) file to accommodate international conventions such as Date (-d) format, Number (-E) format, Character Set, Collation, and Case preferences. For more information on international conventions, see the *Progress Internationalization Guide*.

23 ♦ Choose the Character Set, Collation, Case option. The Language, Country, CharSet, Collation, Case dialog box appears:



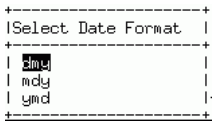
NOTE: You might need to use the down arrow key to reach the end of the list.

If you do not choose an item from this menu, the following error message appears when you continue the installation:

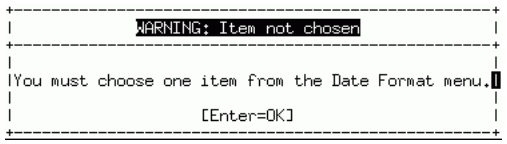


Press **RETURN**, and complete the required setting.

24 ♦ Choose the Date Format option. The Select Date Format dialog box appears:

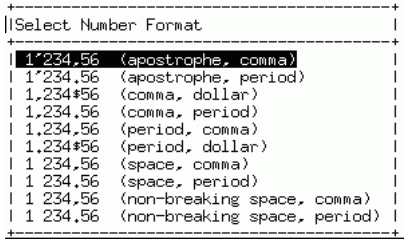


If you do not choose an item from this menu, the following error message appears when you continue the installation:

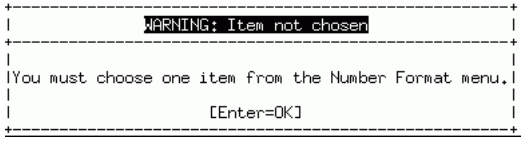


Press **RETURN**, and complete the required setting.

25 ♦ Choose the Number Format option. The Select Number Format dialog box appears:

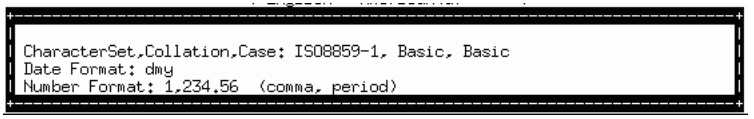


If you do not choose an item from this menu, the following error message appears when you continue the installation:



Press **RETURN**, and complete the required setting.

26 ♦ As you select options in the International Settings dialog box, the Status bar at the bottom of the display shows your choices. Choose the Continue with Installation option, when the Status bar shows a value for each of the settings:



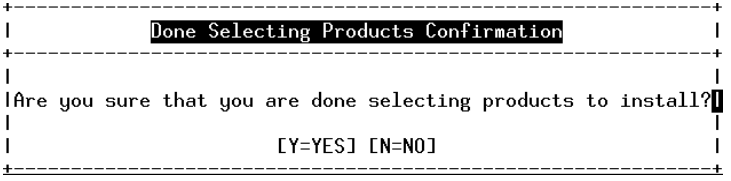
27 ♦ Once you provide all required international settings, the installation performs a check of available disk space. If you have insufficient disk space, a message appears:



28 ♦ Press **RETURN**, and change the path. Provided your new destination choice has enough disk space, the Summary dialog box appears, displaying a summary of the installation options you selected:

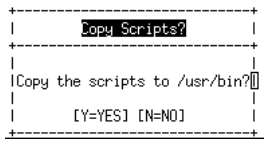


29 ♦ Press **RETURN**, and complete the required setting. The following confirmation dialog box appears:



30 ♦ Type **Y** to continue with the installation or **N** to select additional products.

The Copy Scripts dialog box appears:



WebSpeed consists of several scripts and program modules. The installation utility places them into the default installation directory `/usr/dlc/bin`. This dialog box allows you to place copies into `/usr/bin`.

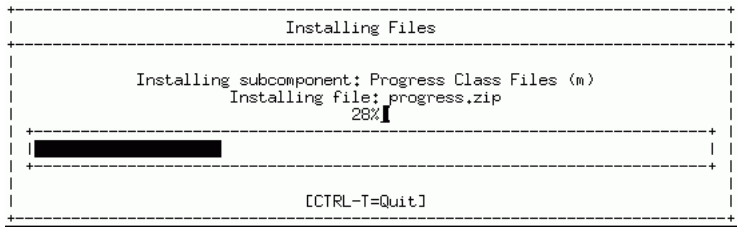
31 ♦ Type **Y** to instruct the installation utility to place WebSpeed scripts in `/usr/bin` and your installation directory.

Type **N** to instruct the installation utility to place WebSpeed scripts **only** in the default directory.

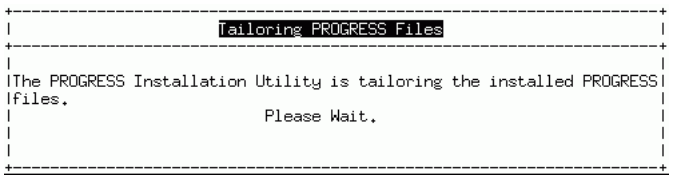
CAUTION: If you have other WebSpeed or Progress products already installed on this machine, answer **No** when prompted to copy the scripts to `/usr/bin`. If you answer **Yes** the installation utility overwrites any existing Progress scripts in `/usr/bin`.

NOTE: If you are maintaining two versions of WebSpeed or Progress on the same machine, answer yes to this question only if most users will be using the version you are now installing.

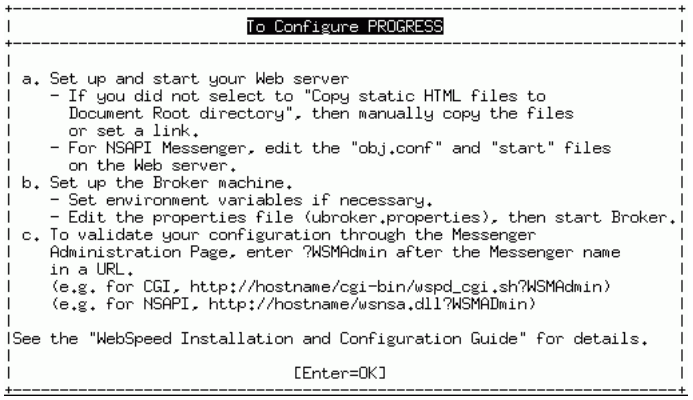
While the utility is decompressing the files, the following dialog box appears:



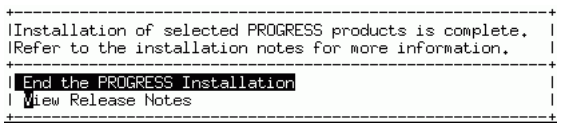
While the utility is tailoring the files, the following dialog box appears:



If you are installing a WebSpeed product such as the Messenger component, the To Configure Progress dialog box appears:



32 ♦ Press **RETURN**. The following dialog box appears:



33 ♦ Choose End the Progress Installation to exit. Progress returns you to the UNIX system prompt.

34 ♦ You can now unmount the CD by entering the following command:

```
umount device-name mount-point
```

You are now ready to verify the installation by configuring and running the WebSpeed Broker and running the test procedure. For instructions, see Chapter 8, “Configuring WebSpeed On UNIX.”

6.4 Running the Installation Utility In Batch Mode

This section describes running the Progress installation utility in batch mode.

6.4.1 Installation Initialization File

Running the Progress installation utility in batch mode requires an initialization file in addition to the other Progress installation support files. You can create an installation initialization file using any text editor. Typically, you would save the initialization file with a `.ini` extension. The following shows the format of the an installation initialization file:

Sample Installation Initialization File *(1 of 3)*

```
Configuration Count]
NumberofConfigurations=n

[Product Configuration 1]
name=Company_name
serial=serial_number
version=version_number
control=control_number_1 control_number_2 control_number_3
prodname=Product_name
.
.
.
[Product Configuration n]
name=Company_name
serial=serial_number
version=version_number
control=control_number_1 control_number_2 control_number_3
prodname=Product_name
[Language Choice]
lang=Arabic
lang=Chinese (Simple)
lang=Chinese (Traditional)
lang=Czech
lang=Danish
lang=Dutch
lang=English - American
lang=English - International
lang=Finnish
lang=French
lang=German
lang=Greek
lang=Hebrew
lang=Hungarian
lang=Icelandic
```

Sample Installation Initialization File *(2 of 3)*

```
lang=Italian
lang=Japanese
lang=Korean
lang=Norwegian
lang=Persian
lang=Polish
lang=Portuguese
lang=Portuguese - Brazilian
lang=Russian
lang=Serbian
lang=Spanish
lang=Spanish - Latin
lang=Spanish - Mexican
lang=Swedish
lang=Thai
lang=Turkish

[Language Default]
_szDefaultLanguage=English - American

[International Settings]
; See intlsets.txt on your CDROM for acceptable values
cpinternal=ISO8859-1
cpcollation=Basic
cpcase=Basic
dateformat=mdy
numsep=46
numdec=44

[Messenger]
InstallingWhereWebServerIsInstalled=yes


[WebServer]
; Type NSAPI
; Type CGI (If the type is CGI then the CgiScriptPath MUST exist)
WebServerType=CGI
CgiScriptPath=directory-pathname/silentCgiScriptDir
CopyHtmlFiles=yes
DocumentRootPath=directory-pathname/silentDocRootDir
```

Sample Installation Initialization File *(3 of 3)*

```
[Java]
jdkHomePath=/<jdk-path>

[Server Engine]
; 4GL
; SQL
; BOTH
type=BOTH

[Type and Destination]
; Type is COMPLETE or TYPICAL
type=TYPICAL
path=directory-pathname/dlcsilent
workpath=directory-pathname/wrk
```

6.4.2 Running a Batch Installation

The syntax for running the Progress Installation utility in batch mode follows:

```
<directory_pathname>/proinst
-b <path>/<filename.ini> -l <path>/<logfile.log>
```

In the syntax above, *directory pathname* refers to the directory where the Progress Installation utility, the installation support files, and the archive files can be found.

For example, a typical batch installation command might be:

```
proinst -b /test/install.ini -l /log/test.log
```

If no installation initialization (`.ini`) file is specified as the argument to the **-b** option, the Progress Installation utility searches the current directory for the default installation file, `install.ini`. If no install initialization file is found, the batch installation fails.

All error messages are redirected to a log file. You can specify a log filename as an argument to the **-l** option. If no filename is specified for the log file, errors are redirected to the default log filename `install.log`. If no directory is specified for the log file, the installation utility checks the TMP, TEMP, and TMPDIR environment variables. It writes the log file to the first valid directory that it finds defined in those variables.

6.5 Setting AdminServer Security

Once you have installed WebSpeed on UNIX, you can optionally set user and/or group AdminServer security. You can set this option on the command line to require an individual user and/or groups of users to provide valid values during the AdminServer startup process. These products use the AdminServer: 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 installation process.

Using the Command Line In UNIX Platforms To Set the Admin Server User-Group Administration Feature

To implement the User-Group Authorization feature on a UNIX platform, you must first successfully complete the installation program.

Table 6–8 identifies and briefly describes the purpose of each new command-line option:

Table 6–8: User-Group Parameter Options

Parameter name...	Syntax...	Purpose
Individual user name and password required	-requireusername	To require a minimum of one user ID to be resolved for each AdminServer operation before it can be executed.
Group authorization required	-admingroup group [{,|:}group...]	To require a minimum of one group to be resolved for each AdminServer operation before it can be executed. On a UNIX platform, a comma-separated list differentiates groups when you are specifying multiple groups on the command line.

On UNIX platforms, a group name can be any user-defined or NIS group name. UNIX can also support subgroups.

7

Configuring WebSpeed On Windows

This chapter describes how you can configure WebSpeed to run your transaction-based Web applications. It also describes administrative features available with WebSpeed. You should be familiar with the WebSpeed run-time architecture before you configure the WebSpeed environment and run a WebSpeed application. See Chapter 2, "Overview Of the WebSpeed Architecture," for more information.

The chapter covers the following topics:

- WebSpeed Configuration Overview
- WebSpeed Administration
- Setting Up WebSpeed On the Web Server Machine
- Placing Static Files On the Web Server Machine
- Overview Of the ubroker.properties File
- Starting the AdminServer
- Setting Up the WebSpeed Environment
- Configuring WebSpeed Components
- Starting WebSpeed Transaction Server and NameServer Instances
- Maintaining the WebSpeed Transaction Server and NameServer Log Files
- Configuring a WebSpeed Messenger-only Installation

- Starting WebSpeed To Test the Configuration
- Testing Your Configuration
- Managing the WebSpeed Transaction Server
- Managing the WebSpeed Messenger

7.1 WebSpeed Configuration Overview

You must perform the following preliminary tasks before you can begin configuring the WebSpeed installation:

- Install the necessary WebSpeed components. For more information about the distribution of resources in the WebSpeed environment, see Chapter 3, "WebSpeed Configurations."
- Configure the machines involved in the WebSpeed installation. This includes setting the appropriate environment variables and setting up your Web server. For more information about configuring the rest of the your WebSpeed environment, see Chapter 4, "Before You Install WebSpeed."

Once you complete these preliminary tasks, you can begin to configure the WebSpeed components. The following overall steps take you through this process:

1 ♦ Start the AdminServer process on each machine. (See the "Starting the AdminServer" section.)

2 ♦ Once the AdminServer is running, you can use the Progress Explorer to create and modify NameServer and WebSpeed Transaction Server configurations. Progress stores these configurations in the properties file (`ubroker.properties`) on the machine where you installed the WebSpeed Transaction Server. (See the "Progress Explorer" section.)

You can also edit the `ubroker.properties` file manually by using a text editor. If you choose to use a text editor, you must have file system access to the file. (See the "Overview Of the ubroker.properties File" section.)

You can validate edited NameServer and WebSpeed configurations using the validation utilities (NSCONFIG and WSCONFIG). These utilities must have access to the properties file. (See the "WebSpeed Command-line Utilities" section.)

3 ♦ Determine if you need to set (or change any preset) WebSpeed environment variables. (See the "Setting Up the WebSpeed Environment" section.)

4 ♦ Use the Progress Explorer (or the NSMAN utility) to start up a NameServer instance that you configured to coordinate client access to one or more WebSpeed Transaction Server instances on your network. After the NameServer has started, use the Progress Explorer (or the WTBMAN utility) to start up each WebSpeed Transaction Server instance controlled by that NameServer. (See the "Starting WebSpeed Transaction Server and NameServer Instances"section.)

After a WebSpeed Transaction Server starts up, it registers its location and the Application Services it supports with its controlling NameServer.

5 ♦ Set up the WebSpeed Messenger on your Web server. (See the "Setting Up WebSpeed On the Web Server Machine" section.)

6 ♦ At any time after Step 5, you can verify that the AdminServer, the NameServer, and the WebSpeed Transaction Server are running and test the configuration to confirm that you have set up WebSpeed correctly. You perform this verification with the Progress Explorer, the WebSpeed command-line utilities, or the WebSpeed Messenger Administration (WSMAdmin) page in a browser. (See the "Testing Your Configuration" section for more information.)

7 ♦ You can shut down the AdminServer process at any time on a WebSpeed or NameServer machine. If you shut down the AdminServer while there are NameServer and WebSpeed Transaction Server instances still running, those instances are shut down as well.

7.2 WebSpeed Administration

The WebSpeed administration framework consists of the following system administration components:

- The AdminServer, which provides access to Progress servers to be administered on the local machine
- The Progress Explorer graphical interface tool, which allows local and remote administration and configuration of WebSpeed and other Progress components
- The management utilities, which allow administration from the command line of WebSpeed and other Progress components

This framework provides a consistent structure for all the WebSpeed Version 3.1 and Progress Version 9.1 server products installed on your network.

7.2.1 The AdminServer

The AdminServer, the core of the common administration framework, supports the managing of WebSpeed and other Progress products (for example, NameServer, database, DataServer).

The AdminServer runs as a service on Windows NT and UNIX platforms. By default, it starts automatically. If you have altered the default behavior, follow these steps to start the AdminServer:

1 ♦ From the Windows NT desktop, choose Start→ Settings→ Control Panel→ Services.

2 ♦ Select the AdminService for Progress 9.1A, and choose Start.

Alternately, you can run the following command, either from the command prompt or a batch file:

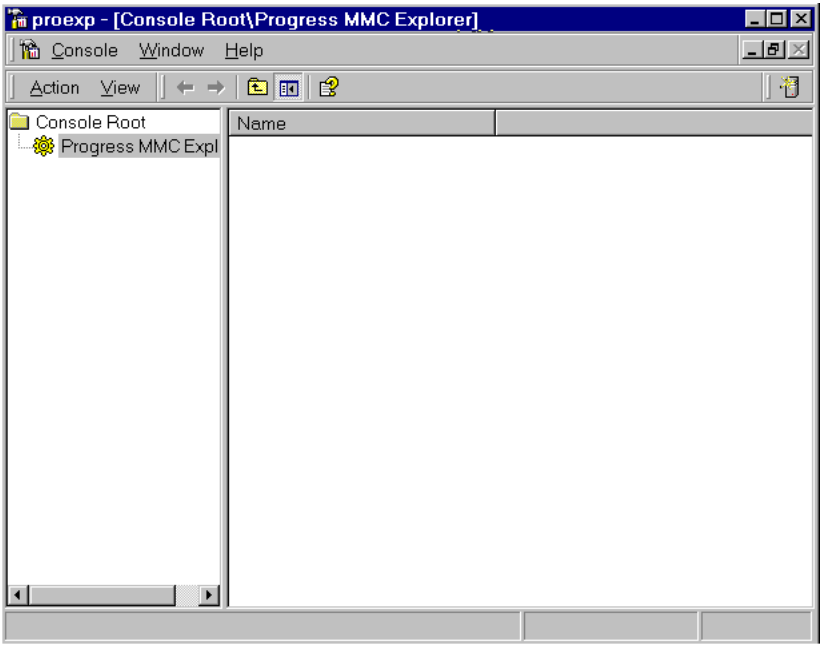
```
NET START ''AdminService for PROGRESS 9.1A''
```

7.2.2 Progress Explorer

The Progress Explorer is a graphical administration utility that operates as a plug-in to the Microsoft Management Console (MMC). The Progress Explorer combines the functionality of all the command-line utilities with the ability to create, save modifications to, and delete individual WebSpeed Transaction Servers, NameServers, DataServers, AppServers, and databases. You can also use the Progress Explorer to configure WebSpeed Messengers, start additional WebSpeed Agents, or trim back running WebSpeed Agents.

See the Progress Explorer's online help for instructions on adding AdminServers to the Progress Explorer and connecting to running AdminServers.

When you start the Progress Explorer, the following window displays:



The interface includes a menu bar and two distinct frames. The left frame presents a tree view that might include WebSpeed Transaction Servers, WebSpeed Messengers, NameServers, AppServers, DataServers, and Progress databases. Expanding any of the folders under an AdminServer shows a list of all configured instances of that component. With a component selected, you can create new instances (except WebSpeed Messengers) by choosing Action→ New. With a component instance selected, you can start, stop, configure, get status views, and read the log files for that instance.

When you install WebSpeed, a sample WebSpeed Transaction Server (`wsbroker1`) and a sample NameServer (`NS1`) are installed automatically. You administer each product instance through a tabbed property sheet, which appears in the Explorer's right frame. You use the tabs to configure or modify all the properties for a specific instance. A toolbar is also available, enabling you to start, save, delete, check the status of, or stop a WebSpeed Broker or NameServer instance.

For information about how to use the Progress Explorer to configure WebSpeed, see the Progress Explorer online help.

7.2.3 WebSpeed Command-line Utilities

The WTBMAN and WSCONFIG command-line utilities provide the ability to manage and query the status of running WebSpeed Transaction Servers and serve as diagnostic tools to display a broker's configuration

WTBMAN Utility

You can use the WTBMAN utility to control the operation of a WebSpeed Transaction Server. The utility allows you to start a Transaction Server, query its status, start and stop additional WebSpeed Agents, trim by a certain number of agents, and shut down the Transaction Server.

For more information about the WTBMAN utility, see the "WTBMAN Utility" section in Appendix C, "Configuration and Management Utilities."

WSCONFIG Utility

You can use the WSCONFIG utility to validate existing WebSpeed Transaction Server or WebSpeed Messenger configurations. The WSCONFIG utility reads the `ubroker.properties` file for validation.

The WSCONFIG configuration command runs locally only, on the machine where the WebSpeed components that you want to check are installed.

NOTE: Because the WSCONFIG utility does not run across the network and no AdminServer is installed during a Messenger-only installation, you cannot use the WSCONFIG utility to check a Messenger-only installation.

For more information about the WSCONFIG utility, see the "WSCONFIG Utility" section in Appendix C, "Configuration and Management Utilities."

7.2.4 NameServer Command-line Utilities

Running as an independent process within the WebSpeed architecture, the NameServer supports load-balancing functionality (when the WebSpeed Enterprise Transaction Server is installed) and location transparency. The NSMAN and NSCONFIG command-line utilities manage, query the running status of, and display the NameServer configuration.

NSMAN Utility

You can use the NSMAN utility to control the operation of a configured NameServer. The utility provides the ability to start a NameServer, query a NameServer status, and shut down a NameServer.

NOTE: The NameServer can simultaneously support WebSpeed, AppServers, and DataServers.

For more information about the NSMAN utility, see the "NSMAN Utility" section in Appendix C, "Configuration and Management Utilities."

NSCONFIG Utility

You can use the NSCONFIG utility to query the current configuration of an existing NameServer and can be used to view all the option values for a specific NameServer. This utility is a diagnostic tool and can be helpful when you are attempting to validate and resolve configuration settings.

The NSCONFIG configuration command runs locally only, on the machine where the NameServer is installed. The utility does not run across the network.

NOTE: The NameServer can simultaneously support WebSpeed, AppServers, and the DataServers.

For more information about the NSCONFIG utility, see the "NSCONFIG Utility" section in Appendix C, "Configuration and Management Utilities."

7.3 Setting Up WebSpeed On the Web Server Machine

This section describes how to set up WebSpeed on your Web server machine. Keep in mind that the machine running your Web server can be the same machine running the Transaction Server. However, you might want to use more than one machine, depending on the size of your WebSpeed application, the computing power of your Web server machine, and security considerations.

NOTE: Since WebSpeed can run on a wide range of Web servers, it is not possible to provide specific instructions here for configuring your Web server. For specific information about your Web server, see your Web server documentation. For a general discussion of preparing your Web server for your WebSpeed installation, see the "Preparing Your Web Server" section in Chapter 4, "Before You Install WebSpeed."

7.3.1 Installing the Messenger Executable

The WebSpeed Messenger must reside on the same machine as your Web server. The Messenger executables are tailored to run with a specific type of Web server. WebSpeed provides a Messenger executable that supports the Web server types described in Table 7–1.

Table 7–1: Windows NT WebSpeed Messengers

Web Server Type	Messenger	Messenger Executable
Microsoft IIS	ISAPI WSASP[1]	wsisa.dll wsasp.dll
Netscape	NSAPI	wsnsa.dll
CGI-compatible	CGI	cgiip.exe

[1] The WSASP Messenger calls WebSpeed applications from an Active Server Page. It cannot coexist with the ISAPI Messenger.

The NSAPI executables reside and run from the *install-path*\bin directory. The CGI Messenger and ISAPI executables reside and run from the \scripts directory on the Web server.

You can use the sample file cgiip.wsc to set up a file association for running the CGIIP Messenger under Microsoft's IIS Server. For details, see the cgiip.wsc file, which is located in the *install-path*\bin directory.

NOTE: You must restart an ISAPI or Netscape NSAPI Web server after installing and configuring the Messenger.

Compatibility With the Web Server Type and Operating System

The Messenger executable comes with the WebSpeed Transaction Server product. You install the Transaction Server on the machine where you want to run the broker and agents.

If you purchased a WebSpeed Transaction Server for an operating system that is different from the operating system that runs your Web server, go to the Download area on Progress Technical Support's Web site at http://www.progress.com/services/support/ and download a compatible Messenger. This is necessary because you must install a Messenger that is compatible with the Web server's operating system. For example, if you are distributing WebSpeed components across networked machines and your Web server is running on UNIX, be sure to install a WebSpeed Messenger executable that is suitable for running on UNIX. Make sure that you download a Messenger for the appropriate UNIX platform and for the appropriate Web server type.

7.3.2 Configuring Netscape Web Servers For NSAPI Messengers

To configure a Web server to work with the WebSpeed NSAPI Messenger, you must edit the Netscape Web server configuration file (`obj.conf`).

Editing the Netscape Web Server Configuration File

A Netscape Web server uses information in its configuration file to recognize the WebSpeed NSAPI Messenger. The configuration file for the Netscape Enterprise Server is named *install-dir*`\https-`*host-name*`\config\obj.conf`. (If you are using the Fast Track Server, see your Web server documentation for the name of the server's configuration file.)

Make a copy of the file before you modify it so that you can restore the original configuration. Table 7–2 describes the changes you must make to `obj.conft`.

Table 7–2: Netscape Web Server Configuration

New Line	Description
`Init fn=load-modules shlib="`*pathname*`" funcs=WSNSAinit,WSNSAdefault, WSNSAshutdown,WSNSAwebspeedCheck`	This pathname is the absolute path to the Messenger DLL, `wsnsa.dll`. The Init line must appear **after** any existing Init commands. This line informs the Web server that it must load the Messenger DLL at the end of the initialization process, and that the named functions are external entry points within the DLL.
`Init fn=WSNSAinit`	This Init line must appear as the last Init command. It informs the Web server that the named function is an external entry point within the DLL.
`NameTrans fn=WSNSAwebspeedCheck`	This NameTrans line must appear **before** any existing NameTrans commands. You must add it within the block of commands delimited by <Object name=default> and </Object>. This line informs the Web server to handle WebSpeed requests properly.
`Service method=(GET|POST|HEAD) fn=WSNSAdefault`	This Service line must appear **before** any existing Service commands. You must add it within the block of commands delimited by <Object name=default> and </Object>. This line informs the Web server to handle WebSpeed requests properly.

Each line you add to `obj.conf` must be on a single line. Do not add line breaks within a command line. Use forward slashes (/) in pathnames. Here is an excerpt from a sample `obj.conf` file (the additions that you must make for the WebSpeed Messenger are bold):

```
Init ...
Init ...
# The following directive is a single line; it contains no line breaks
Init fn=load-modules shlib="c:/Program Files/Progress/bin/wsnsa.dll"
     funcs=WSNSAinit,WSNSAdefault,WSNSAshutdown,WSNSAwebspeedCheck
Init fn=WSNSAinit
<Object name=default>
AuthTrans ...
AuthTrans ...
NameTrans fn=WSNSAwebspeedCheck
NameTrans ...
NameTrans ...
PathCheck ...
PathCheck ...
ObjectType ...
ObjectType ...
Service method=(GET|POST|HEAD) fn=WSNSAdefault
Service ...
Service ...
AddLog ...
AddLog ...
Error ...
</Object>...
...
...
...
```

Applying the Configuration Changes

Access the Netscape Server's browser-based Admin panel and apply the configuration changes before restarting the Web server.

7.3.3 Restarting the Netscape Web Server

After editing the Netscape Web server configuration file (`obj.conf`) to support the WebSpeed NSAPI messenger, you must restart the Web server so that it recognizes this newly installed Messenger.

Before running any WebSpeed application, make sure that your Web server is up and running. Consult your Web server documentation for more information about getting the Web server fully up and running.

Where To Place the Messenger Executable File

Typically, a Messenger script file, such as `cgiip.exe`, resides in the `\scripts` or equivalent directory that contains your Web server's scripts. When you configure your Web server, you can decide which directories can hold executable files.

Most Web servers map URLs leading with `/scripts` to a `\scripts` subdirectory. This subdirectory is located either under or parallel to the document root directory. The `\scripts` directory typically contains only executable files. This is an appropriate location to place your Messenger script file.

7.4 Placing Static Files On the Web Server Machine

You typically use WebSpeed to create part of a larger overall Web presentation. Some of the HTML pages that you present to the Web user never change throughout the life of your presentation. These types of pages are called *static* HTML pages; they are never processed by WebSpeed. Java class files that your Web application uses are also considered static files. WebSpeed Workshop and WebSpeed applications rely on the static HTML and Java class files in *install-path*`\webspeed`. Static Web files must be located under your Web server's document root directory. Alternately, you can configure your Web server to look for static files in the directory of your choice.

If your WebSpeed application uses static HTML files, you should place them on the Web server machine in a subdirectory of the Web server's document root directory (or consult your Web server documentation for information about other options, such as creating an additional document root directory). You must place image files (`.gif` or `.jpeg`), audio files (`.au`), and video files (`.mpeg`, `.mpg`) on the Web server machine. These files cannot be served directly by WebSpeed. You must place these files on the Web server machine.

WebSpeed objects or compiled code must be accessible and visible to the WebSpeed Agents. Make the files accessible by placing them on the machine where your Transaction Server runs. You can make a file visible to the WebSpeed Agents by adding its directory name to the Transaction Server's PROPATH.

7.5 Overview Of the ubroker.properties File

The ubroker.properties file is the property file for all Unified Broker products, WebSpeed Messengers, and the NameServer. All values that define instances of the WebSpeed Transaction Server and the NameServer are stored within this file. The Progress Explorer and the command-line utilities use this information when working with instances of all processes.

The ubroker.properties file resides in the *install-path*\dlc\properties directory. It is a fully commented file containing information relevant to setting properties for your WebSpeed configuration.

From a Windows 95, Windows 98, or Windows NT 4.0 machine, you use the Progress Explorer to create and configure instances of the WebSpeed Transaction Server or the NameServer on the Windows NT platform or remote UNIX platforms. It is possible to edit do this by editing the ubroker.properties file manually. See the "Overview Of the ubroker.properties File" section in Chapter 8, "Configuring WebSpeed On UNIX," for more information on the ubroker.properties file.

7.6 Starting the AdminServer

The AdminServer provides a command and control interface for the WebSpeed Transaction Server, NameServer, and other components. As described earlier in this chapter, the AdminServer must start before you can start WebSpeed.

The AdminServer runs as a service on Windows NT. By default, it starts automatically. If you have altered the default behavior, follow these steps to start the AdminServer:

1 ♦ From the Windows NT desktop, choose Start→ Settings→ Control Panel→ Services.

2 ♦ Select the AdminService for Progress 9.1A, and choose Start.

Alternately, you can run the following command, either from the command prompt or a batch file:

```
NET START ''AdminService for PROGRESS 9.1A''
```

7.7 Setting Up the WebSpeed Environment

Once you complete the WebSpeed installation, you must configure it to your particular requirements. You can manage the configuration of WebSpeed on the Windows NT platform by using the Progress Explorer graphical user interface configuration tool or the WebSpeed command-line utilities.

7.7.1 Setting Environment Variables

Environment variables required by WebSpeed are set during installation in the `ubroker.properties` file. You might need to set the PROPATH and other standard environment variables (for example, DLC) on the WebSpeed Transaction Server machine.

NOTE: In distributed configurations, you must edit the appropriate environment variables on each machine where you have WebSpeed components installed.

You can change most of these settings using the Progress Explorer or by editing the WebSpeed property file, `ubroker.properties`. Note that it is not necessary to modify the Windows NT registry or the system environment variables (through the NT Control Panel).

PROPATH and Other Standard Progress Environment Variables

When you install the WebSpeed Transaction Server, the installation process sets PROPATH for you in the `ubroker.properties` file. PROPATH initially includes a number of subdirectories in your installation directory. In addition, the PROPATH includes a dot (.) directory reference. When the agent sees the dot, the process substitutes the name of its current working directory. For example, the agents resolve the dot to their broker's default directory, which is the working directory.

You can override installed PROPATH settings using the PROPATH property in the properties file (`ubroker.properties`).

Working Directory Settings

The properties file relies on a default setting for the working directory that you specify during installation. You can remove or modify the references in the properties file to establish your own working directory settings for both the WebSpeed Transaction Server and the NameServer.

For more information on Progress environment settings, see the "Environment Variables" section in the "Setting Up Your Environment" chapter of the *Progress Installation and Configuration Guide Version 9 for Windows*.

Configuring WebSpeed and NameServer Log Files

For disk management reasons, you might want to specify a non-default location for the log files used by WebSpeed. A WebSpeed installation uses a number of different log files, which are stored in the default working directory. For example:

- **`[TransactionServername.]server.log`**—WebSpeed Transaction Server log file set using the srvrLogFile property in the [Ubroker] section of the properties file
- **`[TransactionServername.]broker.log`**—WebSpeed Broker log file set using the brokerLogFile property in the [Ubroker] section of the properties file
- **`[NameServername.]ns.log`**—NameServer log file set using the srvrLogFile property in the [NameServer] section of the properties file

After you decide where you want 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 WebSpeed Components" section.

Because log files receive the WebSpeed and NameServer startup and shutdown messages, Progress system messages, and trace messages, the file can grow quickly. If you have the Append option set in the Transaction Server's configuration, these log files do not truncate automatically. In this case, you should periodically trim the file with a text editor. You might want to archive the file contents as you do it. For more information on maintaining log files, see the "Maintaining the WebSpeed Transaction Server and NameServer Log Files" section.

7.8 Configuring WebSpeed Components

You can use the Progress Explorer to create and configure the NameServer and the WebSpeed Broker. You can use Progress Explorer or the command-line utilities to manage the configuration.

For more information about creating, configuring, and managing with the Progress Explorer, see the Explorer online help.

For information about managing with the command-line utilities, see the "WebSpeed Command-line Utilities" section.

7.8.1 Understanding the NameServer's Load Balancing Option

Load balancing is a feature that allows client connection requests to be distributed, based on load, among multiple Unified Broker instances that support the same Application Service. If your product includes load balancing (for example, the WebSpeed Enterprise Transaction Server) or you have installed the load-balancing option for a product that requires it (for example, the AppServer), 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, these weight factors give some sense of the amount of work that an individual WebSpeed Transaction Server instance can handle. For example, Table 7–3 shows the effect of weight factors specified for three WebSpeed Transaction Server instances registered for the same application service.

Table 7–3: Weight Factors Based On Percentage

WebSpeed Transaction Server Name	Weight Factor	% of Time Selected
WS1	20	20
WS2	20	20
WS3	60	60

The selection algorithm used by the NameServer guarantees that WS1 and WS2 are each selected 20% of the time and WS3 is selected 60% of the time. Thus, if the sum of weight factors for all WebSpeed Transaction Server 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 WebSpeed Transaction Server instance over time.

Arbitrary Weight Factors

You can specify any sum of values (not necessarily 100), but the weight of each is always proportional to the sum, as shown in Table 7–4.

Table 7–4: Weight Factors Based On Arbitrary Sums

WebSpeed Transaction Server Name	Weight Factor	% of Time Selected
WS1	2	2/7
WS2	2	2/7
WS3	3	3/7

Fail-over Weight Factor

Another use of Weight Factors is to provide connection-level fault tolerance, by keeping a backup Transaction Server ready to take over the load if the other Transaction Servers in your network fail. If you assign a Weight Factor of zero to the backup Transaction Server, the NameServer does not select it to fill any of the incoming Web requests. If the NameServer loses connection with all the other Transaction Servers that support the application service, the NameServer directs all the Web requests to the backup Transaction Server.

For more information on load balancing and fault tolerance, see the “Configuring Progress Unified Broker Products” chapter in the *Progress Installation and Configuration Guide Version 9 for Windows*.

7.9 Starting WebSpeed Transaction Server and NameServer Instances

The WebSpeed installation includes one sample NameServer and one sample WebSpeed Transaction Server instance. Before you begin to develop or deploy applications on Windows NT, note the following required steps:

1 ♦ The AdminServer must be running. If the AdminServer is not running, you must start it. (See the "Starting the AdminServer" section.)

2 ♦ Start an existing NameServer or create a new NameServer instance. You can create and start a NameServer by using the Progress Explorer; or you can edit the `ubroker.properties` file to create an instance and then use the NSMAN utility to start the instance. When you configure a NameServer instance, you can set it to start up by default whenever the AdminServer starts.

NOTE: The NameServer can be on any machine in your network, even a UNIX machine.

If you are using the Progress Explorer, see the online help for information about creating and starting an instance. If you are editing the `ubroker.properties` file, see the "Editing the ubroker.properties File" section of Chapter 8, "Configuring WebSpeed On UNIX."

To start a local instance of the NameServer from the command line, use the following command:

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

where `NS1` is the name of the NameServer.

To start a remote instance of the NameServer from the command line, use the following command:

```
nsman -name NS1 -host host-name -port port -user user-name -start
```

where *`host-name`* is the name of the host machine on which you want the instance to run; *`port`* is the port number on the AdminServer; and *`user-name`* is the user ID of the system account under which the NameServer will run. If you specify a host name, the Explorer prompts you for a user name (if you do not supply it) and password.

3 ♦ Start an existing WebSpeed Transaction Server or create a new Transaction Server instance. You can create and start a Transaction Server by using the Progress Explorer; or you can edit the `ubroker.properties` file to create an instance and then use the WTBMAN utility to start the instance. When you configure a Transaction Server instance, you can set it to start up by default whenever the AdminServer starts.

See the Progress Explorer online help for information about creating and starting an instance.

To start a local instance of the WebSpeed Transaction Server from the command line, use the following command:

```
wtbman -name wsbroker1 -start
```

where `wsbroker1` is the name of the WebSpeed Transaction Server.

To start a remote instance of the WebSpeed Transaction Server from the command line, use the following command:

```
wtbman -name wsbroker1 -host host-name -port port -user user-name -start
```

where *host-name* is the name of the host machine on which you want the instance to run; *port* is the port number on the AdminServer; and *user-name* is the user ID of the system account under which the Transaction Server will run. If you specify a host name, the Explorer prompts you for a user name (if you do not supply it) and password.

By using either the Progress Explorer or the command-line utilities, you can also stop a NameServer or WebSpeed Transaction Server instance, check its status, and increase or reduce the number of running WebSpeed Agents. For more information, see the Progress Explorer online help and the "Using the Progress Explorer To Check Status," and "Managing the WebSpeed Transaction Server" sections in this chapter.

7.10 Maintaining the WebSpeed Transaction Server and NameServer Log Files

The WebSpeed and NameServer log files include information on when the respective WebSpeed and NameServer processes start up and shut down. These files also include system messages and information up to a level of detail that you specify using the loggingLevel property in the `ubroker.properties` file. In addition, the WebSpeed Broker and Agent log files include all Progress system messages that are not assigned an output destination by the OUTPUT TO KEEP-MESSAGES statement. Thus, the information in these log files can be useful when you perform routine maintenance or troubleshooting.

If you have the Append option set in the Transaction Server's configuration, these log files do not truncate automatically. In this case, you should periodically trim the file with a text editor. You might want to archive the file contents as you do it.

For more information on how to configure the log files for your environment, see the "Configuring WebSpeed and NameServer Log Files" section.

7.11 Configuring a WebSpeed Messenger-only Installation

WebSpeed supports n-tier deployment, enabling flexible network structuring so that you can distribute application logic and processing load among many machines across your distributed network. You can partition and deploy your application, thereby breaking up an application that is large and flexibly reorganizing it to run in a network environment that better suits your needs. The WebSpeed Messenger can be part of this n-tier deployment.

The WebSpeed Messenger must reside on the same machine as the Web server. The Web server and the WebSpeed Messengers need not be on the same machine as the rest of the WebSpeed components. Instead, you can install the Web server and the WebSpeed Messengers together on a different machine if you want.

In this configuration, the Messenger must be able to connect remotely to the machines where the NameServers and AdminServers are installed. To do this, you must configure a remote NameServer. For more information on this, see the "Configuring Progress Unified Broker Products" chapter in the *Progress Installation and Configuration Guide Version 9 for Windows*.

7.12 Starting WebSpeed To Test the Configuration

After you set up the WebSpeed Transaction Server machine and the Web server machine, you are ready to run WebSpeed and test your configuration. Follow these steps:

1 ♦ Make sure the AdminServer is running.

2 ♦ Make sure the NameServer is running.

3 ♦ Start an existing WebSpeed Transaction Server.

For more information, see the "Starting WebSpeed Transaction Server and NameServer Instances" section.

7.13 Testing Your Configuration

You can test your configuration in several ways:

- Use the Progress Explorer Status/Manage tab to view a detailed status.
- Use the `-query` option of the NSMAN and WTBMAN utilities to check the status of components.
- View the Messenger Administration page (WSMAdmin).
- Run the `status.p` procedure from a Web browser.

7.13.1 Using the Progress Explorer To Check Status

You can check the WebSpeed configuration status by using the Status/Manage tab in the Progress Explorer. See the Progress Explorer online help for information.

7.13.2 Using the Query Option To Check Status

You can test your configuration and check its status by running the -query option on the NameServer and the WebSpeed Broker.

To query a local NameServer, use the NSMAN utility, which is installed in *install-path*\bin:

```
nsman -name NameServer-name -query
```

To query a remote NameServer, use this command:

```
nsman -name NameServer-name -host host-name -port port -user user-name -query
```

To query a local WebSpeed broker, use the WTBMAN utility, which is installed in *install-path*\bin:

```
wtbman -name ts-name -query
```

where *ts-name* is the name of the Transaction Server that you want to query. You can have more than one Transaction Server running on a single machine.

To query a remote WebSpeed broker, use this command:

```
wtbman -name ts-name -host host-name -port port -user user-name -query
```

The query reports on the Transaction Server's pool of WebSpeed Agents. For each agent, it lists a process ID, its port number, its status, how many requests it has serviced, when it started, and when its status changed.

These are the agent status types:

AVAILABLE	The agent is available to execute application logic.
BUSY	The agent is actively executing application logic (Web object) for a Web user.
LIMBO	The agent is in a transitional state. If the status persists, it indicates an error condition.
LOCKED	The agent is dedicated to a particular Web browser and is only available to the browser whose application locks it.
STARTING	The broker has launched the agent, but the agent has not yet initialized.

7.13.3 Viewing the Messenger Administration Page

You can test to see if the Web server is fully operational by pointing your Web browser to the WSMAdmin page and following the directions. The WSMAdmin page allows you to perform a round-trip test of your entire WebSpeed configuration or to test each component or leg separately.

See the "Managing the WebSpeed Messenger" section for the URL for the WSMAdmin page.

7.13.4 Running the status.p Procedure

You can run the `status.p` procedure from a Web browser. The `status.p` procedure is installed into the *`install-path`*`\src\web\examples` directory. Running this procedure lets you check that WebSpeed and your Web server are fully operational. If you successfully run this procedure, it means that you configured the WebSpeed run-time environment correctly. If you then have errors running your own application, you can be confident that the problem is specific to your application files.

To run the `status.p` procedure from a Web browser, enter the following URL:

```
http://<hostname>/<script-dir>/<Messenger>/WService=<brokername>/src/web/
examples/status.p
```

7.14 Managing the WebSpeed Transaction Server

The WebSpeed Transaction Server is a background process and, as such, does not present a visual interface to the administrator. However, you can manage the Transaction Server with the WTBMAN utility, and you can validate its properties in the ubroker.properties file with the WSCONFIG utility. The WTBMAN utility commands can run remotely; the WSCONFIG utility commands only run locally.

In addition to allowing you to start, query, or stop the Transaction Server, the WTBMAN utility lets you do the following:

- Start additional agents.
- Trim the number of agents.
- Stop the Transaction Server.
- Display command-line help.

7.14.1 Dynamically Starting Additional Agents

To start additional agents, enter the following command:

```
wtbman -name ts-name -addagents number-to-start
```

where *tsr-name* is the name of the Transaction Server specified in the `ubroker.properties` file and *number-to-start* is the number of additional agents you want to start. The number you specify must not exceed the maxSrvInstance value in the `ubroker.properties` file or your license limit.

7.14.2 Trimming Running Agents

To trim agents, enter the following command:

```
wtbman -name ts-name -trimagents number-to-trim
```

where *ts-name* is the name of the Transaction Server and *number-to-trim* is the number of agents you want to stop.

7.14.3 Stopping the Transaction Server

To stop the broker and all the agents in its pool, enter the following command:

```
wtbman -name ts-name -stop
```

To force an immediate shutdown of the Transaction Server and all its agents, enter the following command:

```
wtbman -kill ts-name
```

7.14.4 Accessing Help On WTBMAN

To get information on WTBMAN syntax and usage, enter the following command:

```
wtbman -help
```

7.15 Managing the WebSpeed Messenger

Sometimes, you might want to check the Messenger's status. For example, the NSAPI Messenger maintains persistent connections to the WebSpeed Brokers, which you might want to break manually. WebSpeed provides an Administrative HTML page for the Messenger that you can access.

If you are running an NSAPI Web server, use the following URL:

```
http://host-name[:port]/wsnsa.dll[/WService=appservicename]?WSMAdmin
```

where *host-name* is the name of the host on which the Messenger is running, *port* is the port that your Web server uses (if different from the default port 80), and *appservice-name* is the name of the application service. For example, the following URL requests the Administration page for the NSAPI Messenger on a host named mars:

```
http://mars/wsnsa.dll/WService=wsbroker1?WSMAdmin
```

If you are running an ISAPI Web server, use the following URL:

```
http://host-name[:port]/scripts/wsisa.dll/[/WService=appservice-name]
?WSMAdmin
```

If you are running a CGI Web server, use the following URL:

```
http://host-name/scripts/cgiip.exe[/WService=appservice-name]?WSMAdmin
```

where *host-name* is the name of your Web server machine, *port* is the port that your Web server uses (if different from the default port 80), *scripts* is your Web server's scripts directory, and *appservice-name* is the name of the application service.

8

Configuring WebSpeed On UNIX

This chapter describes how you can configure WebSpeed to run your transaction-based Web applications. It also describes administrative features available with WebSpeed. You should be familiar with the WebSpeed run-time architecture before you configure the WebSpeed environment and run a WebSpeed application. See Chapter 2, "Overview Of the WebSpeed Architecture," for more information.

The chapter covers the following topics:

- WebSpeed Configuration Overview
- WebSpeed Administration
- Setting Up WebSpeed On the Web Server Machine
- Placing Static Files On the Web Server Machinee
- Overview Of the ubroker.properties File
- Starting the AdminServer
- Setting Up the WebSpeed Environment
- Configuring WebSpeed Components
- Starting WebSpeed Transaction Server and NameServer Instances
- Maintaining the WebSpeed Transaction Server and NameServer Log Files
- Configuring a WebSpeed Messenger-only Installation

- Starting WebSpeed To Test the Configuration
- Testing Your Configuration
- Managing the WebSpeed Transaction Server
- Managing the WebSpeed Messenger

8.1 WebSpeed Configuration Overview

You must perform the following preliminary tasks before you can begin configuring the WebSpeed installation:

- Install the necessary WebSpeed components. For more information about the distribution of resources in the WebSpeed environment, see Chapter 3, "WebSpeed Configurations."
- Configure the machines involved in the WebSpeed installation. This includes setting the appropriate environment variables and setting up your Web server. For more information about configuring the rest of the your WebSpeed environment, see Chapter 4, "Before You Install WebSpeed."

Once you complete these preliminary tasks, you can begin to configure the WebSpeed components. The following overall steps take you through this process:

1 ♦ Start the AdminServer process on each machine by using the PROADSV utility. (See the "Starting the AdminServer" section.)

2 ♦ Once the AdminServer is running, you can then use the Progress Explorer or a text editor to create and modify NameServer and WebSpeed Transaction Server configurations. Progress stores these configurations in the properties file (`ubroker.properties`) on the machine where you installed the WebSpeed Transaction Server. You can use the Progress Explorer remotely from a Windows 95, Windows 98 or Windows NT 4.0 machine to access configurations installed on a UNIX machine.

   You can also edit the `ubroker.properties` file manually using a text editor. If you choose to use a text editor, you must have file system access to the file. (See the "Overview Of the ubroker.properties File" section.)

   You can validate edited NameServer and WebSpeed configurations using the validation utilities (NSCONFIG and WSCONFIG). These utilities must have access to the properties file. (See the "WebSpeed Command-line Utilities" section.)

3 ♦ Determine if you need to set (or change any preset) WebSpeed environment variables. (See the "Setting Up the WebSpeed Environment" section.)

4 ♦ Use the NSMAN utility (or the Progress Explorer remotely) to start up a NameServer instance that you configured to coordinate client access to one or more WebSpeed Transaction Server instances on your network. After the NameServer has started, use the WTBMAN utility (or the Progress Explorer remotely) to start up each WebSpeed Transaction Server instance controlled by that NameServer. (See the "Starting WebSpeed Transaction Server and NameServer Instances" section.)

After a WebSpeed Transaction Server starts up, it registers its location and the Application Services it supports with its controlling NameServer.

5 ♦ Set up the WebSpeed Messenger on your Web server. (See the "Setting Up WebSpeed On the Web Server Machine" section.)

6 ♦ At any time after Step 5, you can validate that the AdminServer, the NameServer, and the WebSpeed Transaction Server are running and test the configuration to confirm that you have set up WebSpeed correctly. You perform this validation by using the WebSpeed command-line utilities, the Progress Explorer (remotely, from a Windows 95, Windows 98, or Windows NT 4.0 machine), or the WebSpeed Messenger Administration (WSMAdmin) page in a browser. (See the "Testing Your Configuration" section for more information.)

7 ♦ You can shut down the AdminServer process at any time on the WebSpeed machine. If you shut down the AdminServer while there are NameServer and WebSpeed Transaction Server instances still running, those instances are shut down as well.

8.2 WebSpeed Administration

The WebSpeed administration framework consists of the following system administration utilities:

- The AdminServer, which provides remote administrative access
- The Progress Explorer graphical interface tool, which allows remote WebSpeed administration and configuration from a Windows 95, Windows 98, or Windows NT 4.0 machine
- The management utilities, which allow WebSpeed administration from the command line

This framework allows you to work within a consistent structure for all the WebSpeed Version 3.1 and Progress Version 9.1 products installed on your network.

8.2.1 The AdminServer

The AdminServer, the core of the common administration framework, supports the managing of WebSpeed and other Progress products (for example, NameServer, database, DataServer) and is also used by other Progress processes.

By using the PROADSV utility, you can start up and shut down the AdminServer. Remember that the AdminServer serves as a connection point for both local and remote services for configuration and administration.

8.2.2 Progress Explorer

The Progress Explorer is a graphical administration utility that operates as a plug-in to the Microsoft Management Console (MMC) on Windows 95, Windows 98, and Windows NT 4.0 machines. The Progress Explorer combines the functionality of all the command-line utilities with the ability to create, save modifications to, and delete individual WebSpeed Transaction Servers, NameServers, DataServers, AppServers, and databases. You can also use the Progress Explorer to configure WebSpeed Messengers, start additional WebSpeed Agents, or trim back running WebSpeed Agents.

You can use the Progress Explorer from a remote machine to create, configure, and administer the WebSpeed components on a UNIX machine. For more information on the Progress Explorer, see the Progress Explorer online help.

8.2.3 WebSpeed Command-line Utilities

The WTBMAN and WSCONFIG command-line utilities provide the ability to manage and query the status of running WebSpeed Transaction Servers and serve as diagnostic tools to display a Transaction Server's configuration.

WTBMAN Utility

You can use the WTBMAN utility to control the operation of a WebSpeed Transaction Server. The utility allows you to start a Transaction Server, query its status, start and stop additional WebSpeed Agents, trim by a certain number of agents, and shut down the Transaction Server.

For more information about the WTBMAN utility, see the "WTBMAN Utility" section in Appendix C, "Configuration and Management Utilities."

WSCONFIG Utility

You can use the WSCONFIG utility to validate existing WebSpeed Transaction Server or WebSpeed Messenger configurations. The WSCONFIG utility reads the `ubroker.properties` file for validation.

The WSCONFIG configuration command runs locally only, on the machine where the WebSpeed components that you want to check are installed.

NOTE: Because the WSCONFIG utility does not run across the network and no AdminServer is installed during a Messenger-only installation, you cannot use the WSCONFIG utility to check a Messenger-only installation.

For more information about the WSCONFIG utility, see the "WSCONFIG Utility" section in Appendix C, "Configuration and Management Utilities."

8.2.4 NameServer Command-line Utilities

Running as an independent process within the WebSpeed architecture, the NameServer supports load-balancing functionality (when the WebSpeed Enterprise Transaction Server is installed) and location transparency. The NSMAN and NSCONFIG command-line utilities manage, query the running status of, and display the NameServer configuration.

NSMAN Utility

You can use the NSMAN utility to control the operation of a configured NameServer. The utility provides the ability to start a NameServer, query a NameServer status, and shut down a NameServer.

NOTE: The NameServer can simultaneously support WebSpeed, AppServers, and DataServers.

For more information about the NSMAN utility, see the "NSMAN Utility" section in Appendix C, "Configuration and Management Utilities."

NSCONFIG Utility

You can use the NSCONFIG utility to query the current configuration of an existing NameServer and can be used to view all the option values for a specific NameServer. This utility is a diagnostic tool and can be helpful when you are attempting to validate and resolve configuration settings.

The NSCONFIG configuration command only runs locally, on the machine where the NameServer is installed. The utility does not run across the network.

NOTE: The NameServer can simultaneously support WebSpeed, AppServers, and the DataServers.

For more information about the NSCONFIG utility, see the "NSCONFIG Utility" section in Appendix C, "Configuration and Management Utilities."

8.3 Setting Up WebSpeed On the Web Server Machine

This section describes how to set up WebSpeed on your Web server machine. Keep in mind that the machine running your Web server can be the same machine running the Transaction Server. However, you might want to use more than one machine, depending on the size of your WebSpeed application, the computing power of your Web server machine, and security considerations.

NOTE: Since WebSpeed can run on a wide range of Web servers, it is not possible to provide specific instructions here for configuring your Web server. For specific information about your Web server, see your Web server documentation. For a general discussion of preparing your Web server for your WebSpeed installation, see the "Preparing Your Web Server" section in Chapter 4, "Before You Install WebSpeed."

8.3.1 Installing the Messenger Executable

The WebSpeed Messenger must reside on the same machine as your Web server. The Messenger executables are tailored to run with a specific type of Web server. WebSpeed provides a Messenger executable that supports the Web server types described in Table 8–1.

Table 8–1: UNIX WebSpeed Messengers

Web Server Type	Messenger Executable
NSAPI (Netscape)	`wsnsa.dll`
CGI-compatible	`cgiip`

When you installed WebSpeed, you provided information about your Web server. If you selected CGI, the installation utility installed the CGI Messenger script, `wspd_cgi.sh`, into the directory you specified as your Web server scripts directory. The NSAPI and CGI Messenger executables reside and run from the Progress `install-path/bin` directory.

NOTE: You must restart the Netscape NSAPI Web server after installing and configuring the Messenger.

Compatibility With the Web Server Type and Operating System

The Messenger executable comes with the WebSpeed Transaction Server product. You install the Transaction Server on the machine where you want to run the broker and agents.

If you purchased a WebSpeed Transaction Server for an operating system that is different from the operating system that runs your Web server, go to the Download area on Progress Technical Support's Web site at `http://www.progress.com/services/support/` and download a compatible Messenger. This is necessary because you must install a Messenger that is compatible with the Web server's operating system. For example, if your Web server is CGI-compatible and runs on Windows NT 4.0, you must install a Messenger for a CGI-compatible Web server that can run on NT.

8.3.2 Configuring the Netscape Web Server For the NSAPI Messenger

You must perform the following tasks to configure a Web server to work with the WebSpeed NSAPI Messenger:

- Edit the Netscape Web server configuration file (`obj.conf`).
- Edit the Netscape Web server start file.

Editing the Netscape Web Server Configuration File

A Web server uses information in its configuration file to recognize the WebSpeed NSAPI Messenger. The configuration file for the Netscape Enterprise Server is named `install-dir/https-`*`host-name`*`/config/obj.conf`. (If you are using the Fast Track Server, refer to your Web server documentation for the name of the server's configuration file.)

Make a copy of the file before you modify it so that you can restore the original configuration. Table 8–2 describes the changes you must make to `obj.conf`.

Table 8–2: Netscape Web Server Configuration *(1 of 2)*

New Line	Description
`Init fn=load-modules shlib="`*`pathname`*`" funcs=WSNSAinit,WSNSAdefault, WSNSAshutdown,WSNSAwebspeedCheck`	This pathname is the absolute path to the Messenger shared object, `wsnsa.dll`. The Init line must appear **after** any existing Init commands. This line informs the Web server that it must load the Messenger shared object at the end of the initialization process, and that the named functions are external entry points within the shared object.
`Init fn=WSNSAinit`	This Init line must appear as the last Init command. It informs the Web server that the named function is an external entry point within the shared object.
`NameTrans fn=WSNSAwebspeedCheck`	This NameTrans line must appear **before** any existing NameTrans commands. You must add it within the block of commands delimited by <Object name=default> and </Object>. This line informs the Web server to handle WebSpeed requests properly.

Table 8–2: Netscape Web Server Configuration *(2 of 2)*

New Line	Description
`Service method=(GET|POST|HEAD) fn=WSNSAdefault`	This Service line must appear **before** any existing Service commands. You must add it within the block of commands delimited by <Object name=default> and </Object>. This line informs the Web server to handle WebSpeed requests properly.

Each line you add to `obj.conf` must be on a single line. Do not add line breaks within a command line. Use forward slashes (/) in pathnames. Here is an excerpt from a sample `obj.conf` file (the additions that you must make for the WebSpeed Messenger are bold):

```
Init ...
Init ...
# The following directive is a single line; it contains no line breaks
Init fn=load-modules shlib="/usr/dlc/bin/wsnsa.dll"
     funcs=WSNSAinit,WSNSAdefault,WSNSAshutdown,WSNSAwebspeedCheck
Init fn=WSNSAinit
<Object name=default>
AuthTrans ...
AuthTrans ...
NameTrans fn=WSNSAwebspeedCheck
NameTrans ...
NameTrans ...
PathCheck ...
PathCheck ...
ObjectType ...
ObjectType ...
Service method=(GET|POST|HEAD) fn=WSNSAdefault
Service ...
Service ...
AddLog ...
AddLog ...
Error ...
</Object>...
...
...
...
```

Editing the Netscape Web Server Start File

The Netscape Web server must be able to locate the WebSpeed Messenger and the resources that the Messenger requires for configuration information and messages. Edit the Netscape Web server's start file (for Netscape Web servers, this file is `install-dir/https-`*`hostname`*`/start`) to set the environment variables described in Table 8–3.

Table 8–3: NSAPI Messenger Settings

Environment Variable	Setting
DLC	The WebSpeed Messenger installation path
PROMSGS	The pathname of the PROMSGS file: `/`*`install-path`*`/promsgs`
PROCFG	The pathname of the Progress configuration file: `/`*`install-path`*`/progress.cfg`
WRKDIR	The pathname of your working directory
LD_LIBRARY_PATH	Shared library search path (on Solaris, Compaq Tru64 UNIX and Linux only)
SHLIB_PATH	Shared library search path (on HP-UX only)
LIBPATH[1]	Shared library search path (on AIX only)

[1] On AIX only, you need to add /usr/lib to the shared library search path environment variable.

Here is a sample start file for AIX that includes the necessary WebSpeed information:

```
#! /bin/sh
...
LIBPATH=/usr/lib:$LIBPATH export LIBPATH
...
DLC=/usr/dlc; export DLC
PROMSGS=/$DLC/promsgs; export PROMSGS
PROCFG=$DLC/progress.cfg: export PROCFG
WRKDIR=/usr/workdir export WRKDIR
```

Applying the Configuration Changes

Access the Netscape Server's browser-based Admin panel and apply the configuration changes before restarting the Web server.

8.3.3 Restarting the Netscape Web Server

After editing the Netscape Web server configuration file (`obj.conf`) to support the WebSpeed NSAPI messenger, you must restart the Web server so that it recognizes this newly installed Messenger.

Before running any WebSpeed application, make sure that your Web server is up and running. Consult your Web server documentation for more information about getting the Web server up and running fully.

8.3.4 Configuring a CGI Messenger

Configuring a Messenger to access a CGI-compatible Web server requires only that you create a script file that the Web server uses to invoke the Messenger executable, `cgiip`. You do not have to change the configuration of your Web server. The script file performs these tasks:

- Sets WebSpeed environment variables described in Table 8–4
- Invokes the Messenger executable, `cgiip`

You can find a sample file that you can use as the basis for a shell script to configure the script file at *install-path*`/bin/wspd_cgi.sh`.

Here is an edited version of *install-path*`/bin/wspd_cgi.sh`:

```
# Determine the correct directory where the Messenger
# is installed from either the tailored name or existing value of $DLC.
for what_dlc in "/usr1/pmccorma/wsrt" "$DLC"
do
    [ ! -f "${what_dlc}/bin/cgiip" ] && continue
    DLC=$what_dlc
    export DLC
    break
done
# Set PROMSGS (if not set)
PROMSGS=${PROMSGS-$DLC/promsgs};                export PROMSGS
# Set the user working directory - this is a tailored value
WRKDIR=${WRKDIR-"/usr1/pmccorma/wrk"};                export WRKDIR
# option 1 using host_name and port_num
# $DLC/bin/cgiip pegasus 5001
# option 2 using a service name defined in $DLC/ubroker.properties
$DLC/bin/cgiip -i wsbroker1
# option 3 the "defaultService" defined in $DLC/ubroker.properties
# $DLC/bin/cgiip
# option 4 using a specific properties file name
# $DLC/bin/cgiip -i wsbroker1 -f ./mybroker.properties
# option 5 using a specific properties file name with the "defaultService"
# $DLC/bin/cgiip -f ./mybroker.properties
```

The wspd_cgi.sh script provides and documents additional options for invoking the Messenger executable. You can, for example, specify the default WebSpeed service or use the -f option to specify a named configuration file. Table 8–4 describes the various options.

Table 8–4: Options For Invoking the CGI Messenger

Option	Description
cgiip	The Messenger uses the Transaction Server that you defined as the Default Service in the ubroker.properties file.
cgiip *host-name port-num*	The Messenger uses the Transaction Server whose host name and port number you specify.
cgiip -i *WS-service*	The Messenger uses the Transaction Server service whose name you specify.
cgiip -f *cnf-file*	The Messenger uses the configuration file that you specify. This option overrides the ubroker.properties setting.

NOTE: Specifying a WebSpeed service name in a URL overrides any settings in your Messenger script file.

You can edit the script if you would like to see what information cgiip is sending to the browser for debugging purposes (including HTTP headers). Here is a sample script that shows how you might do this:

```
# save CGI environment
env >/tmp/webapp.env
# save application ouput
outfile=/tmp/webapp.out
/usr/dlc/bin/cgiip -i WSbroker 2>&1 | tee $outfile
```

This script works as follows: After a request from a Web browser initiates this script, the script saves all of the CGI environment information in a file named webapp.env. The script sends its output not only to the browser but also to a file named /tmp/webapp.out. You can look at /tmp/webapp.out to see what was sent to the browser. The technique is useful for pinpointing problems that might be related more to the server than to your application. A clue that this might be the case is if your browser receives no output from your application and instead displays server errors.

Remember that this script executes every time a browser makes a request. Thus, if multiple browsers make multiple simultaneous requests, the agents servicing the requests can overwrite each other's output to the file. For this reason, you should use this technique only when you are debugging your application (typically with a controlled number of agents and browsers).

Where To Place the Messenger Script File

Typically, a Messenger script file, such as `wspd_cgi.sh`, resides in the `cgi-bin` or equivalent directory that contains your Web server's scripts. When you configure your Web server, you can decide which directories can hold executable files. You can also designate whether the executable files require a specific extension (such as `.cgi`) or can use any extension. See your Web server documentation for details on how to configure a scripts directory and CGI programs.

Most UNIX Web servers map URLs leading with `/cgi-bin` to a `cgi-bin` subdirectory. This subdirectory is located either under or parallel to the document root directory. The `cgi-bin` directory typically contains only executable files. This is an appropriate location to place your Messenger script file.

Many UNIX Web servers allow CGI programs to reside in directories other than `cgi-bin`. However, some Web servers might require a specific extension. See your Web server documentation for more information.

8.4 Placing Static Files On the Web Server Machine

You typically use WebSpeed to create part of a larger overall Web presentation. Some of the HTML pages that you present to the Web user never change throughout the life of your presentation. These types of pages are called *static* HTML pages; they are never processed by WebSpeed. Java class files that your Web application uses are also considered static files. WebSpeed Workshop and WebSpeed applications rely on the static HTML and Java class files in `install-path\webspeed`. Static web files must be located under your Web server's document root directory. Alternately, you can configure your Web server to look for static files in the directory of your choice.

If your WebSpeed application uses static HTML files, you should place them on the Web server machine in a subdirectory of the Web server's document root directory (or consult your Web server documentation for information about other options, such as creating an additional document root directory). You must place image files (`.gif`), audio files (`.au`), and video files (`.mpeg`, `.mpg`) on the Web server machine. These files cannot be served directly by WebSpeed. You must place these files on the Web server machine.

WebSpeed objects or compiled code must be accessible and visible to the WebSpeed Agents. Make the files accessible by placing them on the machine where your Transaction Server runs. You can make a file visible to the WebSpeed Agents by adding its directory name to the Transaction Server's PROPATH.

8.5 Overview Of the ubroker.properties File

The `ubroker.properties` file is the property file for the WebSpeed Transaction Server, WebSpeed Messengers, and the NameServer. All values that define instances of the WebSpeed Transaction Server and the NameServer are stored within this file. The command-line utilities and the Progress Explorer use this information when working with instances of all processes.

The `ubroker.properties` file resides in the `install-path`/properties directory. It is a fully commented file containing information relevant to setting properties for your WebSpeed configuration.

NOTE: The AppServer and the DataServers also use the `ubroker.properties` file to store configuration data. For the purposes of this guide, the `ubroker.properties` file focus is on the WebSpeed Transaction Server and the NameServer. See the appropriate Progress manual for details about viewing and editing configurations applicable to the other products.

8.5.1 The ubroker.properties File Structure

The `ubroker.properties` 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. When configuring your WebSpeed environment, you work most often with the `[UBroker]`, `[UBroker.WS]`, `[NameServer]`, `[WebSpeed]`, and `[WebSpeed.Messengers]` configuration entities.

For more information on the structure of ubroker.properties file, see the "Configuring Progress Unified Broker Products" chapter in the *Progress Installation and Configuration Guide Version 9 for UNIX* and the comments in the file.

8.5.2 Editing the ubroker.properties File

You can edit `ubroker.properties` directly using any text editor to create new WebSpeed Transaction Server and NameServer configurations or edit existing configurations. The simplest way to make new configurations in the `ubroker.properties` file is to copy an existing Transaction Server or NameServer definition and then modify the values of the copy's properties to suit your needs. When you do this, you must be sure to supply each definition with its own uuid setting, as described in the list of required unique parameters later in this section.

From a Windows 95, Windows 98, or Windows NT 4.0 machine, you can also use the Progress Explorer remotely to create and configure instances of the WebSpeed Transaction Server or the NameServer on the UNIX platform.

If you edit the configuration by hand, without the Progress Explorer, note that:

- The `ubroker.properties` file resides in the /`properties` subdirectory of the Progress installation directory.
- You should not directly change the values in the `ubroker.properties` file unless you have a complete understanding of how the changes affect WebSpeed components. If you have the Progress Explorer available from a remote Windows machine, use it to make all changes to this file on your UNIX machines.
- For complete definitions of all the properties and detailed information on how to set them, see the comments included in the installed file.
- If you create additional instances of the WebSpeed Transaction Server and the NameServer, you must be sure that each of the following parameters has a value unique to the entire `ubroker.properties` file:
  - **[Ubroker.WS.broker-name]**—The Transaction Server name must be unique.
  - **portNumber**—Each Transaction Server configuration requires a unique port number.
  - **defaultService**—If you configure two WebSpeed Transaction Servers to use the same NameServer and specify that the Transaction Servers perform the same application service, the Transaction Servers must also support the same business function. Set only one of the Transaction Servers as the default.
  - **appserviceNameList**—A list of aliases that can be used in the WService=<appservice-name> in the URL.

- **uuid**—A universally unique identifier for a Transaction Server. If you use the Progress Explorer to create the new Transaction Server, this property is automatically set. If you manually add Transaction Server definitions, generate a unique uuid for each Transaction Server definition by using the following command:

```
install-path\bin\genuuid
```

  You can then cut and paste the value after "uuid=".

- **Log files**—Each Transaction Server configuration and NameServer configuration generates its own log files; the names must be unique for each Transaction Server and each NameServer.

- If you create additional instances of the WebSpeed Transaction Server and the NameServer by copying an existing instance, be sure that each of the following parameters has the correct values for the new instance:

  - **srvrStartupParam**—Identify the startup parameters for your agents. Copy the value from the `ubroker.properties` file's [UBroker.WS] section to your new Transaction Server definition, and modify.

  - **controllingNameServer**—Make sure that each Transaction Server points to its controlling NameServer.

  - **userName** and **groupName**—You can optionally specify a username and a group name that the Transaction Server runs under; if you do not specify these names, the Transaction Server runs under the username and group name of the user who starts the AdminServer.

NOTE: If you install the NameServer product on a separate host from the WebSpeed Transaction Server, the NameServer installation includes its own copy of the properties file.

You must ensure that all related properties and sections of the file are properly specified for each Transaction Server or NameServer instance. If you do edit the file directly, use the appropriate configuration utility to validate the product configuration that you have edited:

- NSCONFIG for NameServer configurations
- WSCONFIG for WebSpeed Transaction Server configurations

For a complete overview of all the sections in the `ubroker.properties` file and more information on how to edit the file, see the comments in the file or sections on the `ubroker.properties` file in the "Configuring Progress Unified Broker Products" chapter in the *Progress Installation and Configuration Guide Version 9 for UNIX*. For more information on the configuration utilities, see Appendix C, "Configuration and Management Utilities."

8.6 Starting the AdminServer

The AdminServer provides a command and control interface for the WebSpeed Transaction Server, NameServer, and other components. As described earlier in this chapter, you must start the AdminServer before you can start WebSpeed.

The PROADSV utility provides you with the ability to start up and shut down the AdminServer on UNIX. This command runs locally on the AdminServer machine.

Start the AdminServer by running this command:

```
proadsv -start
```

8.7 Setting Up the WebSpeed Environment

Once you complete the WebSpeed installation, you must configure it to your particular requirements. You can manage the configuration of WebSpeed on the UNIX platform by using the WebSpeed command-line utilities.

Alternately, if you are configuring WebSpeed remotely from a Windows machine, you can use the Progress Explorer graphical user interface configuration tool.

8.7.1 Setting Environment Variables

Environment variables required by WebSpeed are set during installation in the `ubroker.properties` file. You might need to set the PROPATH and other standard environment variables (for example, DLC) on the WebSpeed Transaction Server machine.

NOTE: In distributed configurations, you must edit the appropriate environment variables on each machine where you have WebSpeed components installed.

You can change most of these settings using the Progress Explorer (remotely, from a Windows machine) or by editing the WebSpeed property file, `ubroker.properties`.

PROPATH and Other Standard Progress Environment Variables

When you install the WebSpeed Transaction Server, the installation process sets PROPATH for you in the `ubroker.properties` file. PROPATH initially includes a number of subdirectories in your installation directory. In addition, the PROPATH includes a dot (.) directory reference. When the agent sees the dot, the process substitutes the name of its current working directory. For example, the agents resolve the dot to their broker's default directory, which is the working directory.

You can modify this and other environment variable settings in the Progress shell scripts installed on UNIX. You can override installed PROPATH settings using the PROPATH property in the properties file (`ubroker.properties`).

Working Directory Settings

The properties file relies on a default environment variable setting, stored in the WRKDIR environment variable, for the working directory that you specify during installation. You can remove or modify the references in the properties file to establish your own working directory settings for both the WebSpeed Transaction Server and the NameServer.

For more information on Progress environment settings, see the "Environment Variables" section in *Pocket WebSpeed*.

Configuring WebSpeed and NameServer Log Files

For disk management reasons, you might want to specify a nondefault location for the log files used by WebSpeed. A WebSpeed installation uses a number of different log files, which are stored in the default working directory. For example:

- **`[TransactionServername.]server.log`**—WebSpeed Transaction Server log file set using the srvrLogFile property in the [Ubroker] section of the properties file
- **`[TransactionServername.]broker.log`**—WebSpeed Broker log file set using the brokerLogFile property in the [Ubroker] section of the properties file
- **`[NameServername.]ns.log`**—NameServer log file set using the srvrLogFile property in the [NameServer] section of the properties file

After you decide where you want 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 WebSpeed Components" section.

Because log files receive the WebSpeed and NameServer startup and shutdown messages, Progress system messages, and trace messages, the file can grow quickly. If you have the Append option set in the Transaction Server's configuration, these log files do not truncate automatically. In this case, you should periodically trim the file with a text editor. You might want to archive the file contents as you do it. For more information on maintaining log files, see the "Maintaining the WebSpeed Transaction Server and NameServer Log Files" section.

8.8 Configuring WebSpeed Components

You can use the Progress Explorer (remotely, from a Windows 95, Windows 98, or Windows NT 4.0 machine) to configure the NameServer and the WebSpeed Transaction Server. Alternately, you can use the command-line utilities to manage the configuration.

For more information about creating, configuring, and managing with the Progress Explorer, see the Explorer online help.

For information about managing with the command-line utilities, see the "WebSpeed Command-line Utilities" section.

8.8.1 Understanding the NameServer's Load Balancing Option

Load balancing is a feature that allows client connection requests to be distributed, based on load, among multiple Unified Broker instances that support the same Application Service. If your product includes load balancing (for example, the WebSpeed Enterprise Transaction Server) or you have installed the load-balancing option for a product that requires it (for example, the AppServer), 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, these weight factors give some sense of the amount of work that an individual WebSpeed Transaction Server instance can handle. For example, Table 8–5 shows the effect of weight factors specified for three WebSpeed Transaction Server instances registered for the same application service.

Table 8–5: Weight Factors Based On Percentage

WebSpeed Transaction Server Name	Weight Factor	% of Time Selected
WS1	20	20
WS2	20	20
WS3	60	60

The selection algorithm used by the NameServer guarantees that WS1 and WS2 are each selected 20% of the time and WS3 is selected 60% of the time. Thus, if the sum of weight factors for all WebSpeed Transaction Server 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 WebSpeed Transaction Server instance over time.

Arbitrary Weight Factors

You can specify any sum of values (not necessarily 100), but the weight of each is always proportional to the sum, as shown in Table 8–6.

Table 8–6: Weight Factors Based On Arbitrary Sums

WebSpeed Transaction Server Name	Weight Factor	% of Time Selected
WS1	2	2/7
WS2	2	2/7
WS3	3	3/7

Fail-over Weight Factor

Another use of Weight Factors is to provide connection-level fault tolerance, by keeping a backup Transaction Server ready to take over the load if the other Transaction Servers in your network fail. If you assign a Weight Factor of zero to the backup Transaction Server, the NameServer does not select it to fill any of the incoming Web requests. If the NameServer loses connection with all the other Transaction Servers that support the application service, the NameServer directs all the Web requests to the backup Transaction Server.

For more information on load balancing and fault tolerance, see the “Configuring Progress Unified Broker Products” chapter in the *Progress Installation and Configuration Guide Version 9 for UNIX*.

8.9 Starting WebSpeed Transaction Server and NameServer Instances

The WebSpeed installation includes one sample NameServer and one sample WebSpeed Transaction Server instance. Before you begin to develop or deploy applications on UNIX, follow these steps:

1 ♦ The AdminServer must be running. If the AdminServer is not running, use the PROADSV command to start it. (See the "Starting the AdminServer" section.)

2 ♦ Start an existing NameServer or create a new NameServer instance. You can create and start a NameServer by using the Progress Explorer, or you can edit the `ubroker.properties` file to create an instance and then use the NSMAN utility to start the instance. When you configure a NameServer instance, you can set it to start up by default whenever the AdminServer starts.

NOTE: The NameServer can be on any machine in your network, even a Windows machine.

If you are using the Progress Explorer, see the online help for information about creating and starting an instance. If you are editing the `ubroker.properties` file, see the "Editing the ubroker.properties File" section.

To start a local instance of the NameServer from the command line, use the following command:

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

where `NS1` is the name of the NameServer.

To start a remote instance of the NameServer from the command line, use the following command:

```
nsman -name NS1 -host host-name -port port -start
```

where *`host-name`* is the name of the host machine on which you want the instance to run; *`port`* is the port number on the AdminServer. The user-name and password in the properties file are used when starting a NameServer in this manner from a UNIX system.

3 ♦ Start an existing WebSpeed Transaction Server or create a new Transaction Server instance. You can create and start a Transaction Server by using the Progress Explorer, or you can edit the `ubroker.properties` file to create an instance and then use the WTBMAN utility to start the instance. When you configure a Transaction Server instance, you can set it to start up by default whenever the AdminServer starts.

If you are using the Progress Explorer, see the online help for information about creating and starting an instance. If you are editing the `ubroker.properties` file, see the "Editing the ubroker.properties File" section.

To start a local instance of the WebSpeed Transaction Server from the command line, use the following command:

```
wtbman -name wsbroker1 -start
```

where `wsbroker1` is the name of the WebSpeed Transaction Server.

To start a remote instance of the WebSpeed Transaction Server from the command line, use the following command:

```
wtbman -name wsbroker1 -host host-name -port port -start
```

where *`host-name`* is the name of the host machine where you want the instance to run; *`port`* is the port number on the AdminServer. The user-name and password in the properties file are used when starting a NameServer in this manner from a UNIX system.

By using either the Progress Explorer or the command-line utilities, you can also stop a NameServer or WebSpeed Transaction Server instance, check its status, and increase or reduce the number of running WebSpeed Agents. For more information, see the Progress Explorer online help and the "Using the Progress Explorer To Check Status" and "Managing the WebSpeed Transaction Server" sections in this chapter.

8.10 Maintaining the WebSpeed Transaction Server and NameServer Log Files

The WebSpeed and NameServer log files include information on when the respective WebSpeed and NameServer processes start up and shut down. These files also include system messages and information up to a level of detail that you specify using the loggingLevel property in the `ubroker.properties` file. In addition, the WebSpeed Broker and Agent log files include all Progress system messages that are not assigned an output destination by the OUTPUT TO KEEP-MESSAGES statement. Thus, the information in these log files can be useful when you perform routine maintenance or troubleshoot.

If you have the Append option set in the Transaction Server's configuration, these log files do not truncate automatically. In this case, you should periodically trim the file with a text editor. You might want to archive the file contents as you do it.

For more information how to configure the log files for your environment, see the "Configuring WebSpeed and NameServer Log Files" section.

8.11 Configuring a WebSpeed Messenger-only Installation

WebSpeed supports n-tier deployment, enabling flexible network structuring so that you can distribute application logic and processing load among many machines across your distributed network. You can partition and deploy your application, thereby breaking up an application that is large and flexibly reorganizing it to run in a network environment that better suits your needs. The WebSpeed Messenger can be part of this n-tier deployment.

The WebSpeed Messenger must reside on the same machine as the Web server. The Web server and the WebSpeed Messengers need not be on the same machine as the rest of the WebSpeed components. Instead, you can install the Web server and the WebSpeed Messengers together on a different machine if you want.

In this configuration, the Messenger must be able to connect remotely to the machines where the NameServers and AdminServers are installed. To do this, you must configure a remote NameServer. For more information on this, see the "Configuring Progress Unified Broker Products" chapter in the *Progress Installation and Configuration Guide Version 9 for UNIX*.

8.12 Starting WebSpeed To Test the Configuration

After you set up the Transaction Server machine and the Web server machine, you are ready to run WebSpeed and test your configuration. Begin by performing the following steps:

1 ♦ Make sure the AdminServer is running. If the AdminServer is not running, use the PROADSV utility, which is installed in *install-path*/bin:

```
proadsv -start
```

2 ♦ Make sure the NameServer is running. If the NameServer is not running, use the NSMAN utility to start it.

Typically, the sample NameServer (NS1) starts automatically with the AdminServer. You can use this sample NameServer, or you can create a new NameServer by editing the ubroker.properties file; then start the new NameServer with the NSMAN utility.

3 ♦ Start an existing WebSpeed Transaction Server. You can use the sample WebSpeed Transaction Server (wsbroker1), or you can create a new Transaction Server by editing the ubroker.properties file; then start the new Transaction Server with the WTBMAN command.

You can also use the Progress Explorer remotely from a Windows 95, Windows 98, or Windows NT 4.0 machine to start the NameServer or the WebSpeed Transaction Server. See the Progress Explorer online help for information.

8.13 Testing Your Configuration

You can test your configuration in several ways:

- Use the Progress Explorer Status/Manage tab to view a detailed status.
- Use the -query option of the NSMAN, PROADSV, and WTBMAN utilities to check the status of components.
- View the Messenger Administration page (WSMAdmin).
- Run the status.p procedure from a Web browser.

8.13.1 Using the Progress Explorer To Check Status

You can check the WebSpeed configuration status by using the Status/Manage tab in the Progress Explorer. See the Progress Explorer online help for information.

8.13.2 Using the Query Option To Check Status

You can test your configuration and check its status by running the `-query` option on the AdminServer, NameServer, and the WebSpeed Transaction Server.

To query the AdminServer, use the PROADSV utility, which is installed in *install-path*`/bin`:

```
proadsv -query
```

To query a local NameServer, use the NSMAN utility, which is installed in *install-path*`/bin`:

```
nsman -name NameServer-name -query
```

To query a remote NameServer, add the following parameters to the command:

```
nsman -name NameServer-name -host host-name -port port -user user-name -query
```

To query a local WebSpeed Transaction Server, use the WTBMAN utility, which is installed in *install-path*`/bin`:

```
wtbman -name ts-name -query
```

where *ts-name* is the name of the Transaction Server that you want to query. You can have more than one Transaction Server running on a single machine.

To query a remote WebSpeed Transaction Server, add the following parameters to the command:

```
wtbman -name ts-name -host host-name -port port -user user-name -query
```

The query reports on the Transaction Server's pool of WebSpeed Agents. For each agent, it lists a process ID, its port number, its status, how many requests it has serviced, when it started, and when its status changed.

These are the agent status types:

AVAILABLE	The agent is available to execute application logic.
BUSY	The agent is actively executing application logic (Web object) for a Web user.
LIMBO	The agent is in a transitional state. If the status persists, it indicates an error condition.
LOCKED	The agent is dedicated to a particular Web browser and is only available to the browser whose application locks it.
STARTING	The broker has launched the agent, but the agent has not yet initialized.

8.13.3 Viewing the Messenger Administration Page

You can test to see if the Web server is fully operational by pointing your Web browser to the WSMAdmin page and following the directions. The WSMAdmin page allows you to perform a round-trip test of your entire WebSpeed configuration or to test each component or leg separately.

See the "Managing the WebSpeed Messenger" section for the URL for the WSMAdmin page.

8.13.4 Running the status.p Procedure

You can run the `status.p` procedure from a Web browser. The `status.p` procedure is installed into the `DLC/src/web/examples` directory. Running this procedure lets you check that WebSpeed and your Web server are fully operational. If you successfully run this procedure, it means that you configured the WebSpeed run-time environment correctly. If you then have errors running your own application, you can be confident that the problem is specific to your application files.

To run the `status.p` procedure from a Web browser, enter the following URL:

```
http://<hostname>/<script-dir>/<Messenger>/WService=<brokername>/src/web/
examples/status.p
```

8.14 Managing the WebSpeed Transaction Server

The WebSpeed Transaction Server is a background process and, as such, does not present a visual interface to the administrator. However, you can manage the Transaction Server with the WTBMAN utility, and you can validate the Transaction Server with the WSCONFIG utility. The validation reads the file and reports the status of the Transaction Server. The WTBMAN utility commands run remotely; the WSCONFIG utility commands only run locally.

You can also use the Progress Explorer remotely from a Windows 95, Windows 98, or Windows NT 4.0 machine to manage and validate the Transaction Server. (With the Progress Explorer, you can also configure the Transaction Server.) See the Progress Explorer online help for more information.

In addition to allowing you to start, query, or stop the Transaction Server, the WTBMAN utility lets you do the following:

- Start additional agents
- Trim the number of agents
- Stop the Transaction Server
- Display command-line help

8.14.1 Dynamically Starting Additional Agents

To start additional agents, enter the following command:

```
wtbman -name ts-name -addagents number-to-start
```

where *ts-name* is the name of the Transaction Server specified in the `ubroker.properties` file and *number-to-start* is the number of additional agents you want to start. The number you specify must not exceed the maxSrvInstance value in the `ubroker.properties` file or your license limit.

8.14.2 Trimming Agents

To trim agents, enter the following command:

```
wtbman -name ts-name -trimagents number-to-trim
```

where *ts-name* is the name of the Transaction Server and *number-to-trim* is the number of agents you want to stop.

8.14.3 Stopping the Transaction Server

To stop the Transaction Server and all the agents in its pool, enter the following command:

```
wtbman -name ts-name -stop
```

To force an immediate shutdown of the Transaction Server and all its agents, enter the following command:

```
wtbman -kill ts-name
```

8.14.4 Accessing Help On WTBMAN

To get information on WTBMAN syntax and usage, enter the following command:

```
wtbman -help
```

8.15 Managing the WebSpeed Messenger

Sometimes you might want to check the Messenger's status. For example, the NSAPI Messenger maintains persistent connections to the WebSpeed Transaction Servers, which you might want to break manually. WebSpeed provides an Administrative HTML page for the Messenger that you can access with the following URL.

If you are running an NSAPI Web server, use:

```
http://host-name[:port]/wsnsa.dll[/WService=appservicename]?WSMAdmin
```

where *host-name* is the name of the host where the Messenger is running, *port* is the port that your Web server uses (if different from the default port 80), and *appservice-name* is the name of the Transaction Server or NameServer.

For example, the following URL requests the Administration page for the NSAPI Messenger on a host named mars:

```
http://mars/wsnsa.dll/WService=wsbroker1?WSMAdmin
```

If you are running a CGI Web server, use:

```
http://host-name/scripts/cgiip.exe[/WService=appservice-name]?WSMAdmin
```

where *host-name* is the name of your Web server machine, `port` is the port that your Web server uses (if different from the default port 80), *scripts* is your Web server's scripts directory, and *appservice-name* is the name of the application service.

9

WebSpeed Dynamic Code-page Support: How It Works and How To Configure It

This chapter explains how WebSpeed dynamic code-page support works and how to configure it. For information on how to code it, see the *WebSpeed Product Update Bulletin*.

This chapter assumes that the reader is an experienced Webspeed developer familiar with internationalization, code pages, and Unicode. For information on developing WebSpeed applications, see the *WebSpeed Developer's Guide*. For information on internationalization, code page, and Unicode, see the *Progress Internationalization Guide*.

This chapter contains the following sections:

- Introduction
- Basic Browser-Agent Interaction
- How Dynamic Code-page Support Works
- Summary of Code-page Conversions
- Configuring the Agent
- Tasks For the WebSpeed Developer

9.1 Introduction

WebSpeed dynamic code-page support lets a WebSpeed Agent running one code page and a Web browser running a different, but compatible, code page exchange data without corrupting it.

With dynamic code-page support, a single Agent with the CPINTERNAL code page set to UTF-8 (an encoding of Unicode) can handle requests from multiple browsers each running a different code page. This is the approach recommended by Progress Software Corporation.

9.2 Basic Browser-Agent Interaction

In order to understand dynamic code-page support, it is necessary to understand how the Web browser and the WebSpeed Agent interact in a typical WebSpeed configuration. (For information on typical WebSpeed configurations, see Chapter 3, "WebSpeed Configurations," in this book.) For the moment, ignore the Web server, the WebSpeed Messenger, the WebSpeed Broker, and code pages. Assume that a browser and an Agent have established communication.

The browser and Agent interact as follows:

1. The browser emits a Web request.
2. The Web request is received by the Agent.
3. The Agent runs the Web Object (the WebSpeed application).
4. The WebSpeed application creates a Web page.
5. The Agent emits the Web page.
6. The Web page is received by the browser.

The preceding steps repeat.

9.3 How Dynamic Code-page Support Works

In dynamic code-page support, each Web request emitted by the browser indicates, directly or indirectly, the name of the code page it is encoded in. The developer modifies the application to make this happen. Typically, the initial Web page of the application displays a preselected list of languages or nationalities, from which the user selects one.

When a Web request reflecting a response to a language-selection page is received by the Agent, the application:

1. Notes the language or nationality selected
2. Determines the corresponding code page
3. Determines the MIME version (as opposed to the Progress version) of the code-page name

   NOTE: For information on the MIME version and the Progress version of code-page names, see the chapter in the *WebSpeed Product Update Bulletin* on tasks for the developer implementing dynamic code-page support.

4. Converts the MIME version of code-page name to the Progress version
5. Sets a cookie called “wscharset” to the MIME version of the code-page name

   OR

   Sets a query field or a hidden form-field called “wscharset” to the MIME version of the code-page name

When a Web request reflecting a response to a Web page other than a language-selection page is received by the Agent, WebSpeed's `web-disp.p` procedure:

1. Searches for the code-page name in the following places:

   a. In a query field called "wscharset" or a hidden form-field called "wscharset"

   b. In a cookie called "wscharset" (if Step a was unsuccessful)

2. Converts the code-page name from MIME format to Progress format and stores the result in the WEB-CONTEXT handle's HTML-CHARSET attribute (if the search in the previous step was successful.

Then, the application ensures that the MIME version of the code-page name will appear in the Web page the application will create. The technique used depends on whether the Web page is created using static HTML or dynamic HTML, as described in the following bullets:

- If the Web page is created using static HTML, the application should contain a META tag with the MIME version of the code-page name, as in the following example:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset="windows-1250"">
```

- If the Web page is created using dynamic HTML, the application uses the OUTPUT-CONTENT-TYPE() WebSpeed API function to insert the MIME version of the code-page name into the HTML document.

  For more information on this technique, see the chapter on dynamic code-page support developer tasks in the *WebSpeed Product Update Bulletin*.

Once WebSpeed's `web-disp.p` procedure stores the Progress version of the code-page name in HTML-CHARSET, the Agent uses HTML-CHARSET to perform the following code-page conversions:

1. Just after an incoming Web request is received, the Agent converts it from the HTML-CHARSET code page to the Agent's CPINTERNAL code page

2. Just before an outgoing Web page is emitted, the Agent converts it from the Agent's CPINTERNAL code page to the HTML-CHARSET code page

9.4 Summary of Code-page Conversions

This section summarizes the code-page conversions that occur with and without dynamic code-page support.

With dynamic code-page support:

- When a WebSpeed Agent receives a Web request, if the WEB-CONTEXT handle's HTML-CHARSET attribute corresponds to the Progress version of a code-page name, the Agent converts the Web request from the HTML-CHARSET code page to the Agent's CPINTERNAL code page.
- Just before the Agent emits a Web page, if the WEB-CONTEXT handle's HTML-CHARSET attribute is set, the Agent converts the Web page from the Agent's CPINTERNAL code page to the HTML-CHARSET code page.

Without dynamic code-page support:

- When an Agent receives a Web request, it is not converted.
- Just before the Agent emits a Web page, the Agent converts it from the Agent's CPINTERNAL code page to the SESSION:CPSTREAM code page.

9.5 Configuring the Agent

The CPINTERNAL code page of a WebSpeed Agent must be compatible with the code page of any Web request it receives. To accomplish this easily, make the Agent's CPINTERNAL code page UTF-8, which is compatible with virtually all code pages Progress supports. This is the approach recommended by Progress Software Corporation.

For more information, see Chapter 10, "Connecting To a Data Source," in this book.

9.6 Tasks For the WebSpeed Developer

For dynamic code-page support to work, the WebSpeed developer must perform certain preliminary tasks. For complete information on these tasks, see the chapter in the *WebSpeed Product Update Bulletin* on tasks for the developer implementing dynamic code-page support.

10

Connecting To a Data Source

This chapter describes how to connect WebSpeed Agents to a data source for application development or deployment. Specifically, it provides instructions for:

- Connecting To a Progress Database
- Connecting To a Non-Progress Data Source Through a DataServer
- Creating an International Web Site

10.1 Connecting To a Progress Database

WebSpeed supplies a demonstration Progress database called Sports2000. You can use the Sports2000 database to run the sample applications and experiment with your first WebSpeed applications.

In a local configuration, the Progress database resides on the same machine as the WebSpeed Transaction Server. The structure of a remote configuration can vary. Chapter 3, "WebSpeed Configurations," shows some of the possibilities. When you are developing in a client/server environment, you can connect to the database in single-user mode. When you develop WebSpeed applications on a Windows system with the WebSpeed Workshop, you must run the Progress database in a multi-user mode. In the WebSpeed Workshop development environment, several Agents must connect to the database simultaneously: one for the Progress Explorer, another for the WebSpeed WebTools, and one for the AppBuilder.

See the chapter on startup in the *Progress Database Administration Guide and Reference* for more information on running the Progress database server and the parameters that you can specify.

You provide the information that an Agent needs to connect to a database when you configure a WebSpeed Transaction Server and specify the Agent options for its pool of Agents. The WebSpeed Agent is a 4GL client running in batch mode and can accept nearly all the startup parameters a standard client can.

This is a sample srvrStartupParam setting in `ubroker.properties` for the sample Transaction Server, WSbroker1, running on a Windows system:

```
srvrStartupParam=-p web\objects\web-disp.p -cpstream iso8859-1 -weblogerror
```

WebSpeed provides many parameters that allow you to control connections to the database and make adjustments to increase performance. These include parameters for controlling record-buffer size, cursor size, and read-only access. In addition to database connection parameters, WebSpeed provides session parameters that allow you to control the conditions under which it compiles your r-code and Web objects and how to handle code page issues. See the *Progress Startup Command and Parameter Reference* for details on session parameters.

10.1.1 Connecting Programmatically

You can also provide connection information programmatically with the SpeedScript CONNECT statement. You can specify a parameter file that includes connection information with the CONNECT statement. However, you cannot connect to a database in the same procedure that accesses that database. You cannot connect to a database within an embedded SpeedScript program that then accesses the database tables and fields. Instead, you should include the CONNECT statement in a separate procedure (.p) that the embedded SpeedScript calls. This ensures that all Agents receive the information they need to connect to the database.

10.1.2 Connecting To a Local Database

The parameter file that contains information for connecting to a local database must include at least the database name. If the database files do not reside in the Agent's working directory, you must fully qualify the database pathname. If you want the Agent to connect to more than one database at a time, you must use the full database path with the Physical Database Name (`-db`) parameter to specify each additional database. The following parameter file connects an Agent to the Sports2000 demonstration database:

```
srvrStartupParam=-p web\objects\web-disp.p -cpstream iso8859-1 -weblogerror
-db c:\progress\wrk\sports2000
```

By default, this connection uses shared memory. If you want to use a network connection instead, specify the Host Name (-H) and Service Name (-S) parameters (described in Table 10–1) when you connect to the database. The network type defaults to TCP.

10.1.3 Connecting To a Remote Database

When connecting to a remote database, an Agent needs information about the database's physical location. Unlike a local connection, where you can choose not to include them, you must include the parameters described in Table 10–1 for a remote connection.

Table 10–1: Remote Connection Parameters

Parameter	Description
Host Name (-H)	Indicates the name of the machine where the database server resides.
Service Name (-S)	Indicates the name of the available service you are calling. This is the service that the database server uses, not a WebSpeed service. See your `\etc\services` file for a list of available services. Note that the database server does not have to use a named service.

This example of a SrvrStartupParam setting from `ubroker.properties` connects to a remote Sports2000 database:

```
srvrStartupParam=-p web\objects\web-disp.p -cpstream iso8859-1 -weblogerror
-db sports2000 -H pluto -S mysv12
```

The remote database is on the pluto host and the database server uses the service, mysv12.

10.2 Connecting To a Non-Progress Data Source Through a DataServer

A WebSpeed application can access a range of databases through the Progress DataServer architecture. There are native DataServers for accessing the ORACLE DBMS and DB2/400 data on AS/400 systems. The Progress DataServer for ODBC allows applications to access Informix OnLine, Microsoft Access, Microsoft SQL Server, SYBASE, and DB2 data on MVS, NT, RS6000.

The following sections supplement the information provided by the applicable *Progress DataServer Guides*. The sections describe the general tasks that you must perform to install and configure your DataServers in a WebSpeed environment. The sections also provide specific instructions for connecting WebSpeed Agents to the various data sources.

Before developing a WebSpeed application that accesses a non-Progress data source, read your DataServer Guide for information on writing SpeedScript code that gives you the results and behavior you expect for your data source. Some elements of SpeedScript do not behave as documented in the standard WebSpeed documentation when accessing non-Progress data sources. There is also additional syntax that allows you to optimize queries by leveraging the strengths of individual database managers that the standard documentation does not describe in detail.

Typically, integrating a non-Progress data source into your WebSpeed environment involves the following basic steps:

- Planning where the DataServer fits in your WebSpeed hardware architecture
- Installing and setting up the DataServer
- Building a schema holder that contains the data definitions in a format required by WebSpeed applications
- Configuring WebSpeed Agents to connect to the schema holder and data source

Planning where to locate a DataServer is a similar process regardless of which DataServer you use. The other steps, however, vary across DataServers. See the appropriate Progress DataServer guide for instructions.

10.2.1 DataServer and WebSpeed Architecture

You have the option of having a local or a remote DataServer. A local DataServer runs on the same machine as the WebSpeed Agents, that is, on the machine where you install the WebSpeed Transaction Server. A remote DataServer runs on a different machine from the WebSpeed Agents, which could also be the machine where your data source resides. This is not a requirement, however.

Figure 10–1 shows a DataServer that is local to the WebSpeed Transaction Server. The database also resides on this machine.

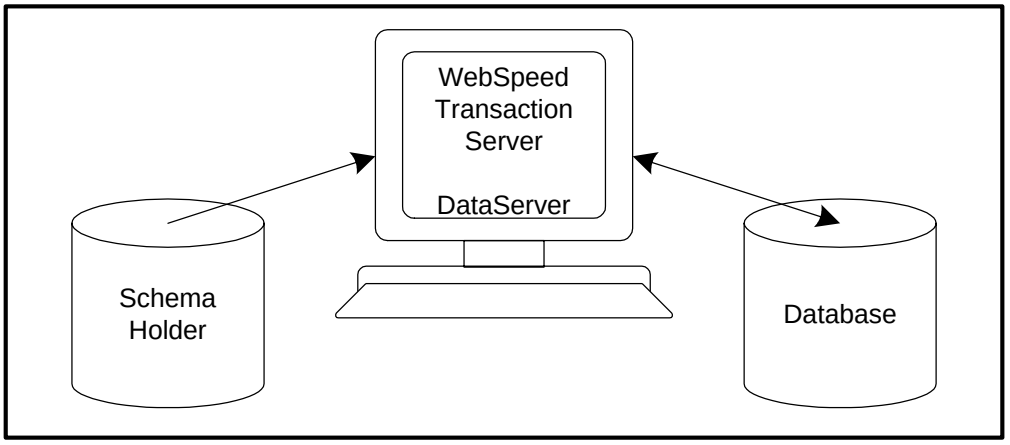


Figure 10–1: The Local DataServer

Figure 10–2 shows one possible configuration for the remote DataServer where a WebSpeed Agent accesses a remote DataServer. Here, the database and the DataServer are running on the same machine. The schema holder is local to the WebSpeed Transaction Server to increase performance.

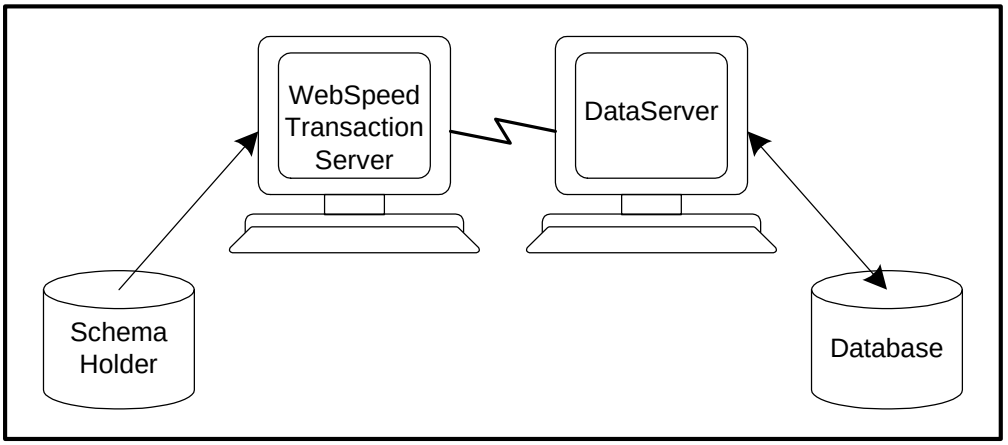


Figure 10–2: Remote DataServer

The figures show a schema holder on the same machine where your Agents run. A local read-only schema holder gives you better performance. However, you might decide that you want your schema holder on a separate machine to make better use of processing resources.

Note that none of the configuration diagrams include a Web server. The Web server plays no role after passing the initial Web request for data to the WebSpeed Messenger. Much like a typical Progress client, the WebSpeed Agent interacts with the DataServer, which in turn passes queries and data requests to the database management system.

You can integrate DataServers for ODBC and ORACLE into the Progress Explorer administrative framework, which allows you to have a single administrative system and take advantage of the NameServer's load balancing. Alternately, the WebSpeed Agents can access DataServers that cannot be managed by the AdminServer, such as the Progress/400 DataServer.

There are variations on these configurations that allow you to combine Windows NT and UNIX platforms or to distribute the components using vendor networking (for example, ORACLE's SQL*Net). The DataServer architecture that allows you to access DB2/400 data is more complex as it is integrated into the native OS/400 architecture. See your *Progress DataServer Guide* for more information. The *DataServer Guides* present configuration information in traditional client/server terms; when reading this material for implementing WebSpeed, substitute WebSpeed Agent for Client.

10.2.2 Setting Up and Running the DataServer

After installing and configuring your DataServer as the *Progress DataServer Guides* describe, follow these general steps:

1 ♦ Start the DataServer processes and create the schema holder. The WebSpeed Transaction Server does not have to be running at this time.

2 ♦ One recommended technique for managing an Agent's connection information is to create a parameter file (.pf). Create a parameter file that specifies the schema holder, the database, user information, and information for accessing a remote database, if your configuration requires it. For more information on creating and using parameter files, see the *Progress Client Deployment Guide*.

3 ♦ Configure the WebSpeed Agent to use the parameter file that you created by specifying the Parameter File (-pf) parameter. For example, if you created a parameter file named `db2dbcon.pf`, your entry in the Startup Parameters field on the Server Parameters tab in the Progress Explorer might be:

```
-p web\objects\web-disp.p -cpstream iso8859-1 -pf db2dbcon.pf
```

4 ♦ Start the Transaction Server.

When the WebSpeed Broker whose Agents you configured with DataServer connection information receives a Web request, it selects an Agent from its pool to service the request by connecting to the non-Progress database and, through the DataServer, retrieving or updating data.

10.3 Creating an International Web Site

If your Web site targets an international audience, consider letting users select the language or nationality they prefer. For each language or nationality, besides translating the text, you can adapt graphics, images, layout, and content. For example, if you are building a Portuguese-language Web site that deals with currency and taxes and that targets users in Portugal and in Brazil, you can adapt the content to deal with Portuguese currency and taxes as well as with Brazilian currency and taxes.

Also, consider using UTF-8 (an encoding of Unicode) for the encoding of the database and for the CPINTERNAL code page of the WebSpeed Agents. This approach is strongly recommend by Progress Software Corporation. For more information on Unicode, see the *Progress Internationalization Guide*.

Also, consider using WebSpeed dynamic code-page support. For more information, see Chapter 9, "WebSpeed Dynamic Code-page Support: How It Works and How To Configure It," in this book.

Figure 10–3 shows a WebSpeed configuration that supports an international Web site.

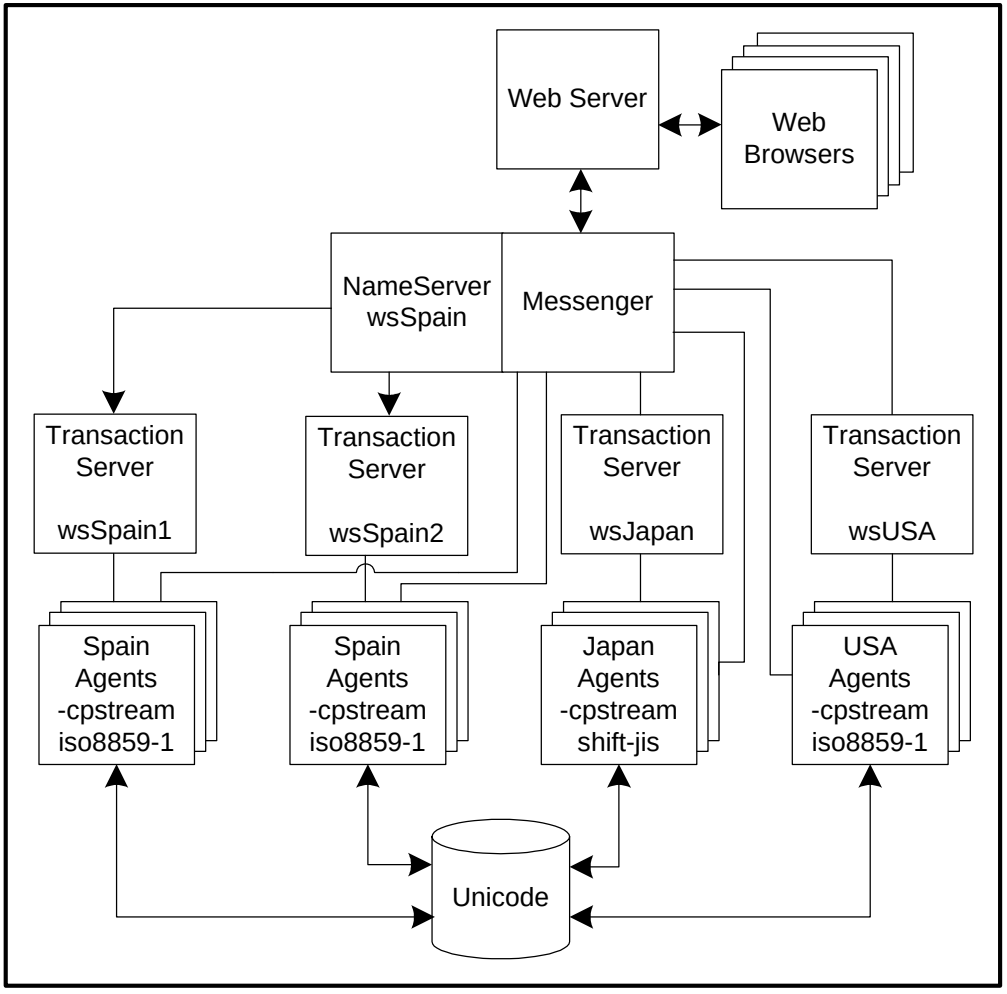


Figure 10–3: WebSpeed Configuration That Supports an International Web Site

This configuration drives a Web site that handles requests from users in Spain, Japan, and the United States. In this example, all countries share a single Unicode (UTF-8) database. This particular Web site expects the highest rate of access from its Spanish users, so it has multiple Transaction Servers and Agent pools configured to handle the Web requests. These Transaction Servers are, in turn, registered with a NameServer. The NameServer can manage the Spanish Transaction Servers as they have access to equivalent resources, that is, their Agents are all configured identically: they use the same internal code page and they connect to the same database in the same manner. Web requests coming from Japanese and U.S. users are each serviced by a single Transaction Server and its Agent pool.

10.3.1 Setting Up Transaction Servers

To set up a Transaction Server to support part of an international WebSpeed application, set up the environment in which it runs and configure its Agent pool to access the appropriate data. To simplify configuration and maintenance, consider creating a subdirectory for each language or nationality. Similarly, consider placing databases, parameter files, and other resources in directories specific to the language or nationality.

When setting up a Transaction Server's environment, you must ensure that when a message is issued by a WebSpeed Agent, the Transaction Server displays it in the appropriate language. To accomplish this, set the PROMSGS variable in the `ubroker.properties` file's Environment section for each Transaction Server definition. You cannot set PROMSGS at a system level, because multiple Transaction Servers requiring different translations of messages will run simultaneously. The international PROMSGS files have a three-letter extension that indicates the language. The following example sets PROMSGS to the Japanese version of the messages file:

```
PROMSGS=/usr/wsrt/prolang/jpn/promsgs.jpn; export PROMSGS
```

Define and configure your Transaction Servers as you would for a standard WebSpeed installation.

If it is important that you have messages displaying in the user's language at all levels of the application, you can use the CGI-compatible Messenger. The NSAPI-compatible Messenger can use only one PROMSGS file, which you configure in its startup script. For the CGI Messenger, you can create a script file (based on `wspd_cgi.sh`) for each language and specify the appropriate PROMSGS file to use.

10.3.2 Configuring Agents

To configure Agents to run in an international environment, you must address code-page issues and session options when you specify Agent parameters. Progress automatically installs the language you select as the default and configures your machine for this default language; however, you must configure each Agent to use the appropriate parameters for the country it accepts requests from. Using a parameter (`.pf`) file helps you manage Agent startup and connection information.

On Windows, use the Progress Explorer to configure the Agents. You add the Parameter File (-pf) startup parameter and name of your parameter file to the Startup Parameters field for the Agent Executable File.

On UNIX, use the Progress Explorer from a remote Windows machine or manually update the `ubroker.properties` file to configure the Agents. You add the Parameter File (-pf) startup parameter and name of your parameter file to the srvrStartupParam option in the section of the `ubroker.properties` file where you define the Transaction Server.

For each supported international language, Progress provides an example parameter file that you can use as the base for creating your own.

The following examples show the contents of parameter files for the Spanish, Japanese, and U.S. Agents, respectively. They are based on the parameter files provided by Progress in the `prolang/lang` directory:

```
-db intldb -cpinternal SHIFT-JIS -cpstream SHIFT-JIS -d ymd
```

```
-db intldb -cpinternal iso8859-1 -cpstream iso8859-1 -d dmy -E
```

```
-db intldb -cpinternal iso8859-1 -cpstream iso8859-1
```

These files provide the information that Agents need to connect to a local database and to run with session options appropriate to their cultural context. For example, each Agent displays dates in a distinct format; in addition, the Spanish Agents use the comma as a radix for decimal data. You can add other parameters to the parameter file to further manage how Agents connect to the databases.

Here is a sample setting for the srvrStartupParam option in the section of the `ubroker.properties` file where you define the wsJapan Transaction Server:

```
srvrStartupParam=-p web/objects/web-disp.p -cpstream iso8859-1 -weblogerror
-p web/objects/web-disp.p -weblogerror -pf japan.pf
```

Alternatively, the WebSpeed application can let the language or nationality be selected by the user. From the selection, the application can set the CPINTERNAL, CPSTREAM, and DATE-FORMAT attributes of the SESSION handle to the appropriate values.

Code Pages

The Agent's CPINTERNAL code page (that is, the code page used by the Agent for its internal processing) must be compatible with the code page of the database it connects to. This is easy to accomplish if the databases's code page is UTF-8. This is the approach recommended by Progress Software Corporation.

The Agent's CPSTEAM code page (which the Stream Code Page (-cpstream) parameter specifies must be the same as the Agent's CPSTREAM code page (assuming dynamic code-page support is not used).

Session Options

Supporting an international Web site means more than just supporting several languages. You must also support the cultural difference in date formats and numeric notations. WebSpeed allows you to control how Agents display and process date information and numeric conventions. The Date (-d) parameter lets you specify the format that an Agent uses to process dates. By default, an Agent processes dates as month, day, year. To display a date as day, month, year, specify -d dmy in the Agent's parameter (`.pf`) file. The European Numeric Format (-E) parameter specifies that a comma (,) represents the decimal point instead of a period (.).

On Windows systems, another aspect of an Agent's session environment that you control through startup parameters is which initialization (`.ini`) file an Agent uses. If you want an Agent to display error messages in a language other than the default language you chose for your WebSpeed installation, you must use an initialization file to set the PROMSGS option to the appropriate version of the PROMSGS file. The WebSpeed Agent can use the [WinChar Startup] section of the initialization file, so you must set PROMSGS there. Specify the Initialization File (-ininame) parameter to specify the name of the initialization file you customized for the Agent.

10.3.3 Directing Web Requests

As you design your international Web site, keep in mind that one of the initial pages should let users choose a specific language or nationality. The URL for each language option should specify the appropriate Transaction Server for the language the user has chosen.

Assuming that the sample configuration shown in Figure 10–3 is running with an NSAPI-compatible Web server, the URLs for the links that Japanese, Spanish, and U.S. users would select are:

```
http://mars/wsnsa.dll/WService=wsJapan/iwebapp
```

```
http://mars/wsnsa.dll/WService=wsUSA/iwebapp
```

```
http://mars/wsnsa.dll/WService=wsSpain/iwebapp
```

The URLs for Japan, Spain, and the U.S. name WebSpeed Transaction Servers to service the Web request. In Figure 10–3 the WebSpeed Transaction Servers are load balanced. Load balancing is available with the WebSpeed Enterprise Transaction Server or as a separately ordered product.

11

Starting Your WebSpeed Configuration

After you have installed and configured WebSpeed on your system, you must start the components in the proper sequence. Chapter 3, "WebSpeed Configurations," shows several sample WebSpeed configurations. This chapter describes how to properly start up those configurations. Specifically, this chapter discusses:

- Startup Options
- Single-machine Configuration
- Development Network With a Central WebSpeed Machine
- Development Network With a Dedicated Web Server
- Multiple LAN Development Environment
- Deployment Configuration With a Dedicated Web Server
- Development and Deployment Shared Configuration

11.1 Startup Options

There are many ways that you can start your WebSpeed configuration. You can configure some or all of your configuration to start automatically when your system starts. You can write scripts to handle starting your configuration. However, because of the dependencies in a WebSpeed environment, there are sequences in which you should generally start the components. For example, the NameServer requires its AdminServer to be running when the NameServer starts.

The following sections describe startup procedures for the sample configurations. The startup procedures avoid common errors that occur when components are started before everything they depend on has started. The routines assume that you are starting everything manually. If you automate the start up of your configuration, use these routines as guides in determining the sequence to start your components.

In most cases, when a machine or component fails, you only have to restart that machine or component. If your data source in a complex distributed configuration fails, data integrity concerns might make it necessary to bring down other components before restarting your data source.

11.2 Single-machine Configuration

Figure 11–1 shows the components installed in a single-machine configuration.

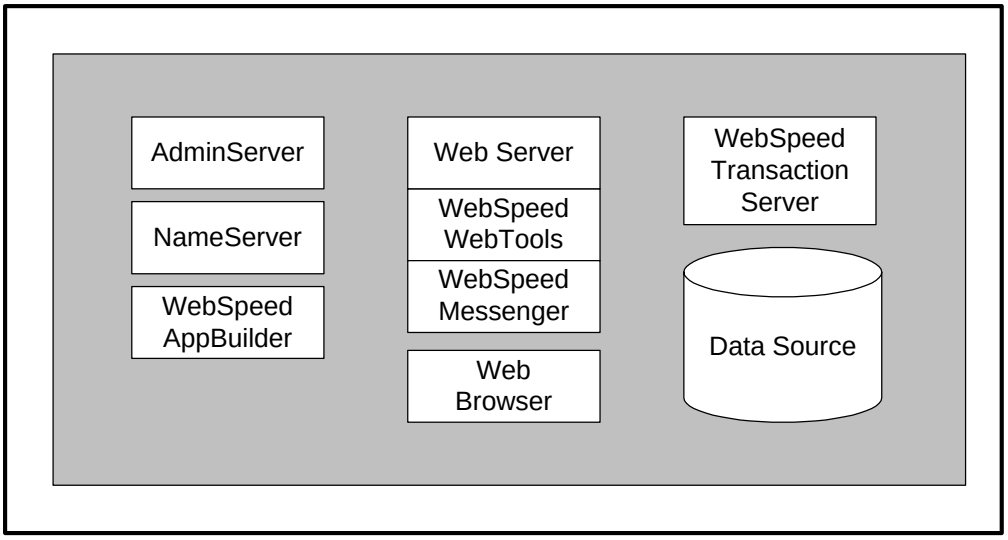


Figure 11–1: Configuration For Single Developer On a Single Machine

Follow these steps to start this configuration:

1 ♦ Start the AdminServer.

NOTE: Once the AdminServer has started, you can use the Progress Explorer to start the other components on this machine.

2 ♦ Start the NameServer.

NOTE: The default setting for the NameServer is to start when the AdminServer starts.

3 ♦ Start the data source in multi-user mode.

4 ♦ Start the WebSpeed Transaction Server.

5 ♦ Start the Web server.

6 ♦ Start the WebSpeed AppBuilder.

7 ♦ Start the Web browser by choosing the WebTools from the AppBuilder's Tools menu as needed.

11.3 Development Network With a Central WebSpeed Machine

Figure 11–2 shows the components installed in a development network with a central machine.

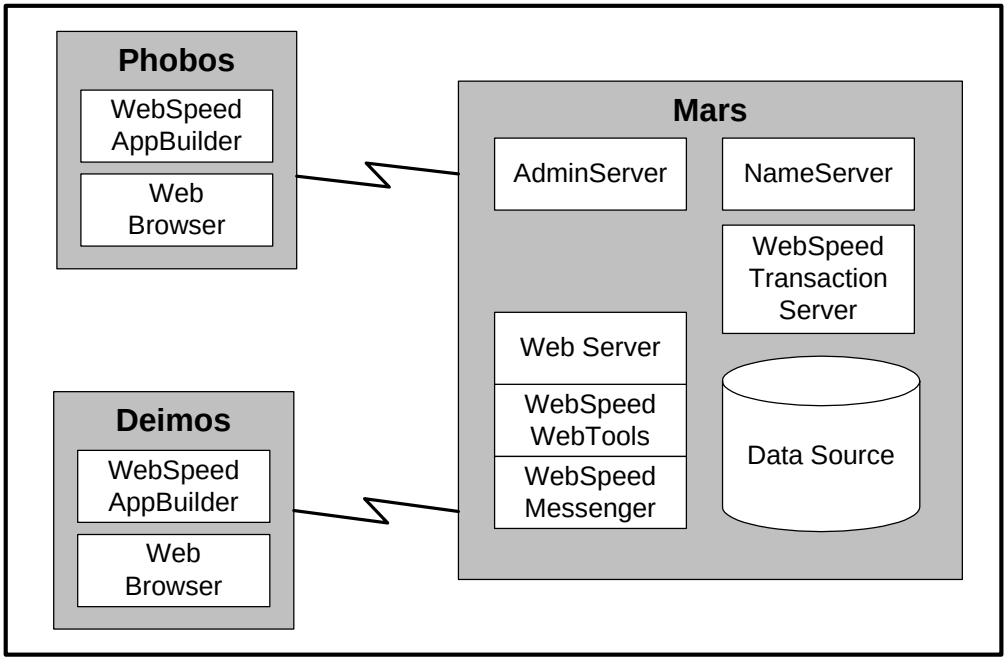


Figure 11–2: Development Network With a Central WebSpeed Machine

Follow these steps to start this configuration:

1 ♦ Start the AdminServer on Mars.

NOTE: Once the AdminServer has started, you can use the Progress Explorer to start other components on Mars.

2 ♦ Start the NameServer on Mars.

3 ♦ Start the data source on Mars in multi-user mode.

4 ♦ Start the WebSpeed Transaction Server on Mars.

5 ♦ Start the Web server on Mars.

6 ♦ Start the WebSpeed AppBuilders on the workstations (Phobos and Deimos) as needed.

7 ♦ Start the Web browser by choosing the WebTools from the AppBuilder's Tools menu on the workstations as needed.

11.4 Development Network With a Dedicated Web Server

Figure 11–3 shows this type of configuration.

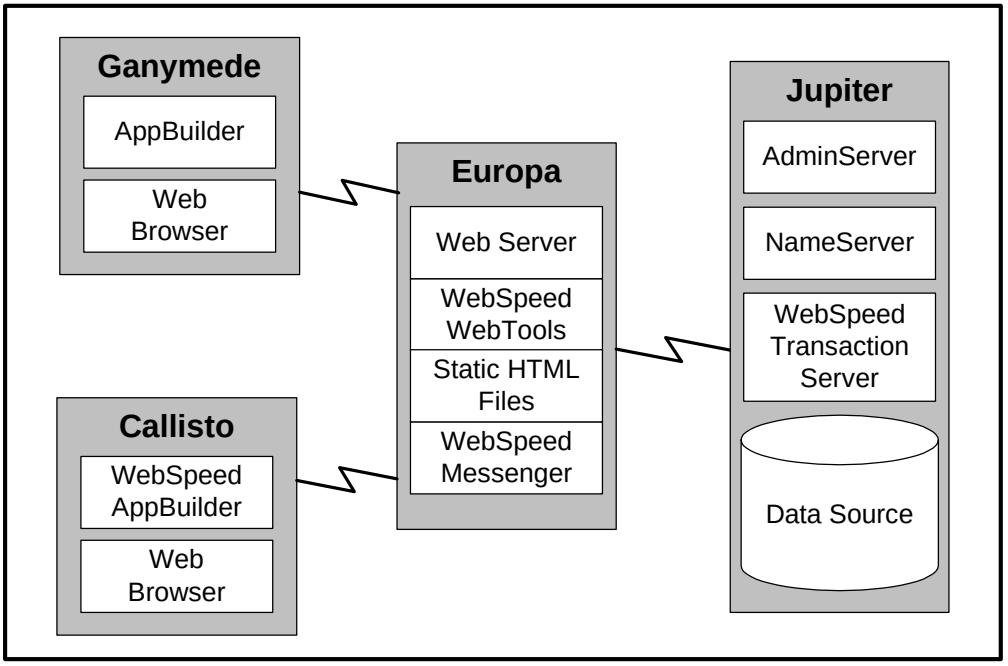


Figure 11–3: Development Network With a Dedicated Web Server

Follow these steps to start this configuration:

1 ♦ Start the AdminServer on Jupiter.

NOTE: Once the AdminServer has started, you can use the Progress Explorer to start other components on Jupiter.

2 ♦ Start the NameServer on Jupiter.

3 ♦ Start the data source on Jupiter in multi-user mode.

4 ♦ Start the WebSpeed Transaction Server on Jupiter.

5 ♦ Start the Web server on Europa.

6 ♦ Start the AppBuilders on the workstations (Ganymede and Callisto) as needed.

7 ♦ Start the Web browsers on the workstations by choosing the WebTools from the AppBuilder's Tools menu as needed.

11.5 Multiple LAN Development Environment

Figure 11–4 shows the products installed on a multiple LAN configuration.

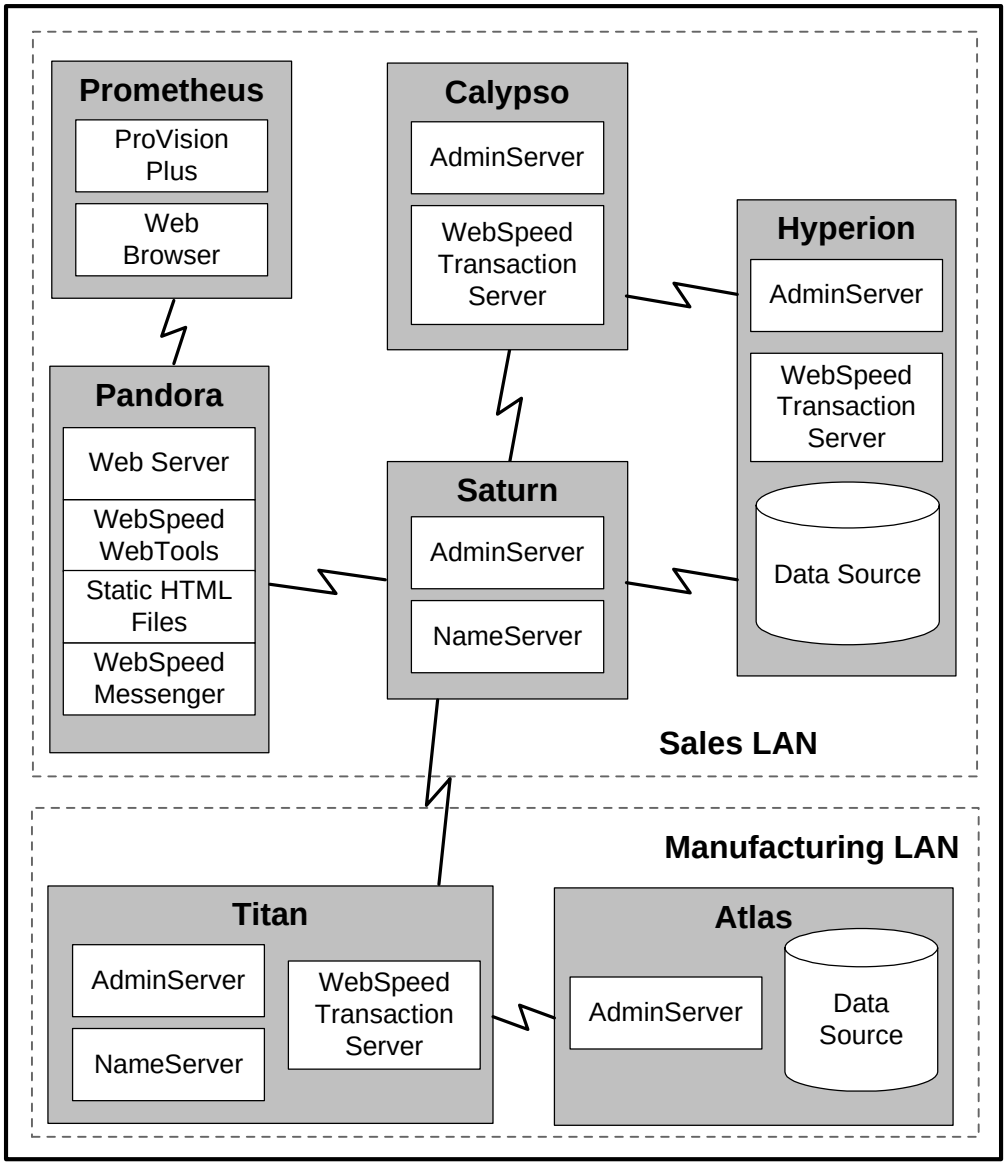


Figure 11–4: Multiple LAN Development Environment

Follow these steps to start this configuration:

1 ♦ Start the AdminServers on Saturn, Calypso, Hyperion, Titan, and Atlas.

NOTE: Once the AdminServers have started, you can use the Progress Explorer from any machine to start other components on these machines.

2 ♦ Start the NameServers on Saturn and Titan.

3 ♦ Start the data sources on Hyperion and Atlas in multi-user mode.

4 ♦ Start the WebSpeed Transaction Servers on Calypso, Hyperion, and Titan.

5 ♦ Start the Web server on Pandora.

6 ♦ Start the AppBuilders on the workstations (like Prometheus) as needed.

7 ♦ Start the Web browsers on the workstations by choosing the WebTools from the AppBuilder's Tools menu as needed.

NOTE: If one of the data sources fails, you should stop all the WebSpeed Transaction Servers that support applications that use that data source. Bring the data source back up and then restart the Transaction Servers. This helps ensure data integrity.

11.6 Deployment Configuration With a Dedicated Web Server

Figure 11–5 shows the products installed in a deployment network with a dedicated web server.

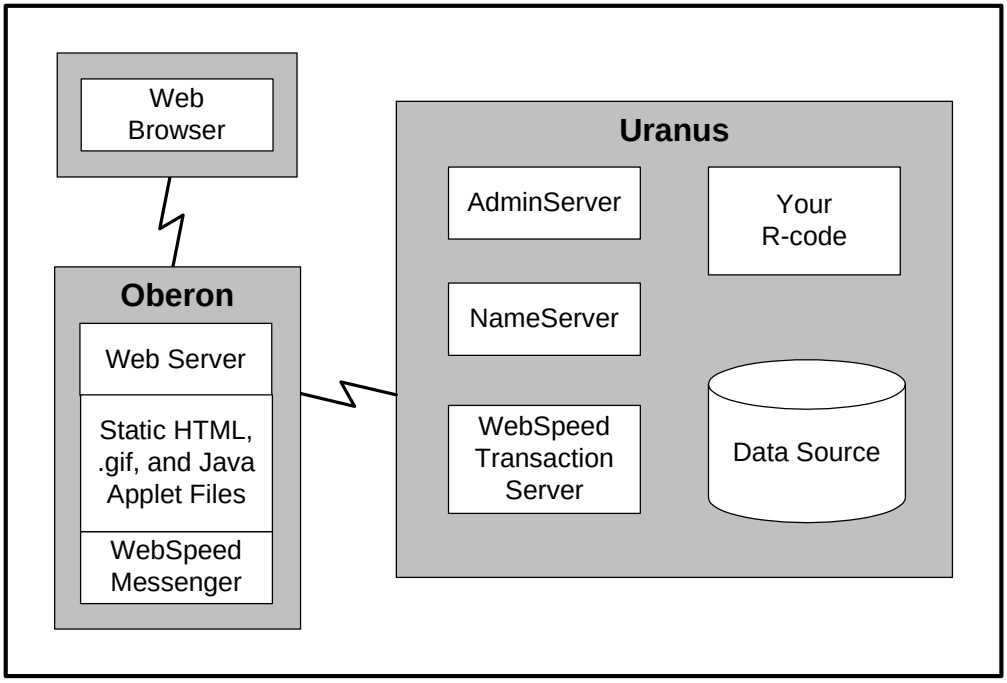


Figure 11–5: Deployment Configuration With a Dedicated Web Server

Follow these steps to start this configuration:

1 ♦ Start the AdminServer on Uranus.

NOTE: Once the AdminServer has started, you can use the Progress Explorer to start other components on Uranus.

2 ♦ Start the NameServer on Uranus.

3 ♦ Start the data source on Uranus in multi-user mode.

4 ♦ Start the WebSpeed Transaction Server on Uranus.

5 ♦ Start the Web server on Oberon.

11.7 Development and Deployment Shared Configuration

Figure 11–6 shows the products installed in a shared development and deployment network.

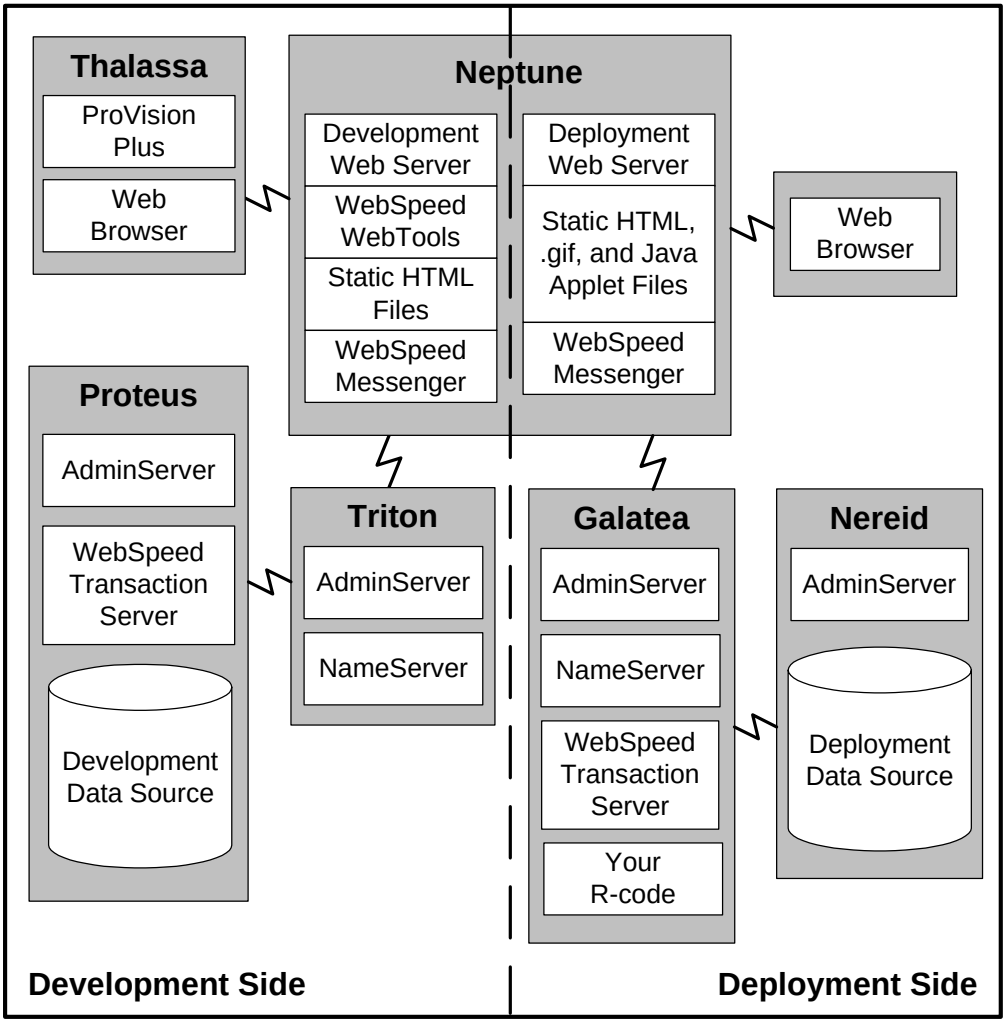


Figure 11–6: Development and Deployment Shared Configuration

Follow these steps to start the Development side of this configuration:

1 ♦ Start the AdminServers on Triton and Proteus.

NOTE: Once the AdminServers have started, you can use the Progress Explorer from any machine to start other components on these machines.

2 ♦ Start the NameServer on Triton.

3 ♦ Start the data source on Proteus in multi-user mode.

4 ♦ Start the WebSpeed Transaction Server on Proteus.

5 ♦ Start the development Web server on Neptune.

6 ♦ Start the AppBuilders on the workstations (like Thalassa) as needed.

7 ♦ Start the Web browsers on the workstations by choosing the WebTools from the AppBuilder's Tools menu as needed.

Follow these steps to start the Deployment side of this configuration:

1 ♦ Start the AdminServers on Galatea and Nereid.

NOTE: Once the AdminServers have started, you can use the Progress Explorer from any machine to start other components on these machines.

2 ♦ Start the NameServer on Galatea.

3 ♦ Start the data source in multi-user mode on Nereid.

4 ♦ Start the WebSpeed Transaction Server on Galatea.

5 ♦ Start the deployment Web server on Neptune.

NOTE: If one of the data sources fails, you should stop all the WebSpeed Transaction Servers supporting applications that use that data source. Bring the data source back up and then restart the Transaction Servers. This helps ensure data integrity.

12

Upgrading WebSpeed On Windows

This chapter describes the conditions and steps required to migrate from WebSpeed Version 2 or Version 3.0 to WebSpeed Version 3.1 on the Windows NT 4.0 operating system. The chapter includes information about the following topics:

- Migrating Your Existing WebSpeed Version 2 Configuration To WebSpeed Version 3.1
- Coexistence Of WebSpeed Version 2 With WebSpeed Version 3.1
- Migrating Your Existing WebSpeed Version 3.0 Configuration To WebSpeed Version 3.1
- Coexistence Of WebSpeed Version 3.0 With WebSpeed Version 3.1

NOTES: WebSpeed provides the SHOWCFG utility, which displays information on the current license. For more information, see the "SHOWCFG Utility" section in Appendix C, "Configuration and Management Utilities."

WebSpeed also provides the Product Update utility, which lets you change the number of users or the expiration date of a Progress product once you have obtained a new License Addendum from your Progress Software sales representative. For more information, see the "Product Update Utility" section in Appendix C, "Configuration and Management Utilities."

12.1 Migrating Your Existing WebSpeed Version 2 Configuration To WebSpeed Version 3.1

Follow these steps to migrate between WebSpeed Version 2 and WebSpeed Version 3.1:

NOTE: Follow these steps before installing WebSpeed Version 3.1 on your system. Be sure to stop all WebSpeed applications before you uninstall WebSpeed Version 2.

1 ♦ Record the current WebSpeed Version 2 configuration with the following command:

```
wsconfig -e filename
```

This command starts the WebSpeed Configuration utility that documents your current WebSpeed settings, except the PROPATH.

2 ♦ The WebSpeed Configuration utility contains an Agent Options tab. The PROPATH settings on this tab are not recorded when you export by WSCONFIG. If you have added values to the PROPATH, write them down. One difference between WebSpeed Version 2 and WebSpeed Version 3.1 is that the PROPATH definition displayed on the Agent Options tab is shared by all Transaction Servers; changing PROPATH in WebSpeed Version 2 affects all Transaction Servers. In WebSpeed Version 3.1, each Transaction Server must be uniquely configured with its own PROPATH.

3 ♦ Save any custom changes made to the Version 2 `web-disp.p` file and re-apply them to the Version 3.1 `web-disp.p` file.

4 ♦ Note any internationalization or code-page issues.

5 ♦ If you have customized your WebSpeed Messenger error messages, save the `wsCusErr.txt` file located in the WebSpeed working directory.

6 ♦ Use the WebSpeed Configuration utility to stop the running Transaction Servers with the following command:

```
wtbman -stop ts-name
```

7 ♦ Reboot the system to make sure that the ISAPI/NSAPI messengers are released from memory.

8 ♦ Uninstall WebSpeed Version 2. Afterwards, delete any remaining Transaction Server session and error log files.

9 ♦ Install WebSpeed Version 3.1 and re-create the Transaction Servers using the Progress Explorer. See Table 12–1 for a list of keywords that have changed from WebSpeed Version 2 to Version 3.1.

As of WebSpeed Version 3.0, the NameServer is a mandatory component. You should review the documentation describing the NameServer. In WebSpeed Version 2, the NameServer is not required and is included only in the Enterprise version. In WebSpeed Version 2, you create the brokers. Then you have the option to create a Dispatcher, and assign brokers to the Dispatcher during configuration. In WebSpeed Version 3.1, you define the NameServer; then, as you create each Transaction Server, you assign the Transaction Server to a NameServer. Remember that you must recompile all 4GL files in your applications.

10 ♦ Start the new WebSpeed Version 3.1 Transaction Servers. See Chapter 7, "Configuring WebSpeed On Windows," for instructions.

Keyword Conversion Chart

Table 12–1 contains a list of WebSpeed Version 2 keywords taken from the Windows NT Registry and associates them with their WebSpeed Version 3.1 equivalents. You can use this chart as a guide during the process of re-creating the WebSpeed Version 2 Transaction Servers and Dispatchers as WebSpeed Version 3.1 Transaction Servers and NameServers. Progress Software recommends that you use the Progress Explorer to re-create the NameServer and WebSpeed Transaction Servers. However, you can manually edit the values used by WebSpeed Version 3.1, which are defined in the `ubroker.properties` file. It is important to note that the location of the keywords within this file is positional; moving keywords between groups could cause errors. If you choose to edit this file manually, see the "Overview Of the ubroker.properties File" section in Chapter 8, "Configuring WebSpeed On UNIX," for more information.

Table 12–1: WebSpeed Version 2 To Version 3.1 Conversion Chart For Windows NT *(1 of 4)*

WebSpeed Version 2 Keywords (NT Registry)	WebSpeed Version 3.1 Keywords (ubroker.properties)
InstallPath—Installation directory; set by the install	Also stored in the registry; set by the install
RootPath—Web server document root directory	RootPath—Web server document root directory

Table 12–1: WebSpeed Version 2 To Version 3.1 Conversion Chart For Windows NT *(2 of 4)*

WebSpeed Version 2 Keywords (NT Registry)	WebSpeed Version 3.1 Keywords (ubroker.properties)
ScriptPath—Web server Messenger directory	ScriptPath—Web server Messenger directory
DefaultService—Transaction Server supports the default application service	defaultService—Transaction Server supports the default application service
AllowMsngrCmds—Allow internal admin commands	AllowMsngrCmds—Allow internal admin commands
Components—Leave at default	components—Leave at default
InfoVersion—Leave at default	infoVersion—Leave at default
WsType—Values 1 for broker, 2 for dispatcher	N/A
PortNum—Transaction Server port number	portNumber—Transaction Server port number
Location—Values 200 for local, 100 for remote	location—local or remote (NameServer only)
StartUp—auto-start value	AutoStart—Yes/No
StartInstances—Initial agents to start	initialSrvrInstance—Initial Agents to start
MaxInstances—Maximum number of agents	maxSrvInstance—Maximum number of Agents
BeginNewInstances—Minimum number of free agents	minSrvrInstance—Minimum number of Agents that will be maintained
AgentMinPort—Minimum port number for an agent process	srvrMinPort—Minimum port number for an Agent
AgentMaxPort—Maximum port number for agent process	srvrMaxPort—Maximum port number for a server process
NumBrokers—Number of brokers controlled by the dispatcher	N/A

Table 12–1: WebSpeed Version 2 To Version 3.1 Conversion Chart For Windows NT *(3 of 4)*

WebSpeed Version 2 Keywords (NT Registry)	WebSpeed Version 3.1 Keywords (ubroker.properties)
IdleTimeout—Wait period to shut down idle agents	IdleTimeout—Time Agent can be idle before Broker flags it to be trimmed if more Agents than minSvrInstance active
PriorityWeight—Transaction Server load balancing weight factor	priorityWeight—Transaction Server load balancing weight factor
HostName—System machine name	hostName—System machine name
DefaultDirectory—Application directory	workDir—Transaction Servers' working directory
ServiceExeFile—Broker executable name	ClassMain—Broker startup executable
WSISADLLFile—ISAPI messenger name	[WebSpeed.Messengers.WSISA] section; set value to msngrExecFile—to the path + executable name
WSNSADLLFile—NSAPI messenger name	[WebSpeed.Messengers.WSNSA] section; set value to msngrExecFile—to the path + executable name
CGIIPExeFile—CGI Messenger name	[WebSpeed.Messengers.CGIIP] section; set value to msngrExecFile—to the path + executable name
IISDllFIle—Renamed broker executable	N/A
AgentExeFile—Agent executable name	srvrExecFile—The agent executable name
AgentParams—Agent startup parameters	srvrStartupParam—Agent startup parameters
ErrorLogFile—Broker error log name	brokerLogFile—Broker error log name
SessionLogFile—Broker session log name	srvrLogFile—Broker session log name
User—The user ID	userName—User ID to start Transaction Server

Table 12–1: WebSpeed Version 2 To Version 3.1 Conversion Chart For Windows NT *(4 of 4)*

WebSpeed Version 2 Keywords (NT Registry)	WebSpeed Version 3.1 Keywords (ubroker.properties)
Group—The group ID for broker process	groupName—Password to start Transaction Server
ServiceNameList—Lists brokers for dispatchers	ServiceNameList—Lists application services for the Transaction Server
Language—Not used	N/A
Environment—Production or development mode	srvrAppMode—Production or development mode
Debugging—Enabled to allow debugging	srvrDebug—Enabled to allow debugging
ApplicationURL—Default is blank	applicationURL—Default is blank
DefaultCookiePath—Default is blank	DefaultCookieDomain—Default is blank
StateTransport—Leave at default	State Transport—Leave at default
StateSavingMethod—Leave at default	StateSavingMethod—Leave at default
StateSavingDirectory—Leave at default	StateSavingDirectory—Leave at default

12.2 Coexistence Of WebSpeed Version 2 With WebSpeed Version 3.1

WebSpeed Version 2 can coexist with WebSpeed Version 3.1. However, issues will arise if you do not perform the installation correctly. Therefore, Progress Software recommends that you do not use mixed configurations.

Because of the way they handle memory, some systems might not let you simultaneously load two versions of the same Messenger into memory. If you must support a mixed configuration, the easiest solution is to set up a different Web server and use a different type of Messenger (CGI, ISAPI, or NSAPI) for each version. You would then have to maintain a copy of your application code for each version, changing the URLs to point to the appropriate Web server and WebSpeed Messenger.

Before you decide whether to maintain both versions, consider the following points:

- **Registry Keys**—WebSpeed Version 3.1 uses a different registry key from the key used by WebSpeed Version 2; therefore, the WebSpeed Version 2 registry will not be overwritten.
- **Program Groups**—WebSpeed Version 3.1 uses a different Program Group than the one used by WebSpeed Version 2, thereby eliminating any potential conflicts.
- **Configuration Mechanisms**—WebSpeed Version 3.1 uses the Progress Explorer, updated command-line utilities, and the `ubroker.properties` file for configuration. Configuration information for WebSpeed Version 2 is stored in the registry.
- **Messengers**—The WebSpeed Version 3.1 Messengers are not intended to be backward-compatible with WebSpeed Version 2 Messengers.

  WebSpeed Active Server Pages (ASP) Messenger coexistence is not supported.

  Configurations using the NSAPI and ISAPI Messenger cannot coexist. Due to the way these Messengers work, an incoming connection cannot be correctly filtered and sent to the appropriate version of the Messenger. Therefore, only one version of these Messengers can exist per Web server. Additionally, it is not possible to configure the Netscape `obj.conf` to support two WebSpeed NSAPI Messengers.

  Because the ISAPI and CGI Messengers are integrated into the directory structure of the Web server, these two messengers are installed into the Web server `\scripts` directory. Because WebSpeed Version 3.1 Messengers use the same file name as the WebSpeed Version 2 Messengers, an installation of WebSpeed Version 3.1 overwrites the previously installed messengers for WebSpeed Version 2.

You can work around this issue by maintaining both sets of messengers. There are two ways that you can achieve this:

– Rename the existing WebSpeed `CGIIP.EXE` and `WSISA.DLL` messengers to a unique `.EXE` and `.DLL` name. Then install WebSpeed Version 3.1 on your system. Both sets of messengers will be in the same Scripts directory. You will then have to maintain two sets of application code, modifying the URLs to reference the new messenger name in one and leaving the other to refer to the original messenger names.

– Modify the Web server by adding an additional virtual directory to be used as a secondary Scripts directory. You can do this through your Web server administration program prior to installing WebSpeed Version 3.1 on your system. During installation, enter the new Scripts directory name when you are prompted for the Web server Scripts directory. Because the location of the messenger has changed, you must make a copy of all your existing WebSpeed applications and modify them so that the application URLs refer to the correct Scripts directory. For example:

```
http://host/scripts/messenger/WService=broker/
yourapplication
```

becomes:

```
http://host/newscripts/messenger/WService=broker/
yourapplication
```

- **System Path**—The System Path presents a possible conflict. A WebSpeed Version 2 user might have added to the System Path a reference to the WebSpeed Version 2 `\bin` directory. WebSpeed Version 2 and WebSpeed Version 3.1 use some of the same command-line utilities, such as WTBMAN. The command-line utilities for Version 3.1 are not backward compatible. If you use the system path to provide access to the WTBMAN utility, only the first WTBMAN found would be executed, and this might be the incorrect version.

  You can work around this by modifying the path specifically for a command shell. You can configure two command shells, each with a unique path and referencing the correct WebSpeed `\bin` directory. If you added the WebSpeed `\bin` directory to the system path, remove it and set it only for the specific command shell.

- **Static HTML Files**—Static HTML files are an issue, especially in development configurations, because both versions of the WebSpeed Workshop use static HTML files and look for them in one specific directory. Because these files have the same names, they cannot reside in the Web server's document root directory. This creates problems for coexistence.

  As a workaround, creating a second Web server on the system, with its own document root directory and running at a different port number, would solve the problem; but, again, it requires changes to the WebSpeed application URLs to incorporate the new port number.

- **Command-line Utilities**—Both WebSpeed Version 2 and WebSpeed Version 3.1 support two command-line utility programs, WTBMAN and WSCONFIG. These utilities are not backward-compatible. WebSpeed Version 3.1 also includes additional command utilities:

  – NSCONFIG is a new tool used to view the current configuration of the NameServer; it functions similarly to WSCONFIG.

  – NSMAN is a new tool used to start, stop, and query the NameServer; it functions similarly to WTBMAN.

12.3 Migrating Your Existing WebSpeed Version 3.0 Configuration To WebSpeed Version 3.1

Follow these steps to migrate between WebSpeed Version 3.0 and WebSpeed Version 3.1:

NOTE: Follow these steps before installing WebSpeed Version 3.1 on your system.

1 ♦ Save any custom changes made to the Version 3.0 `web-disp.p` file and re-apply them to the Version 3.1 `web-disp.p` file.

2 ♦ Note any internationalization or code-page issues.

3 ♦ If you have customized your WebSpeed Messenger error messages, save the `wsCusErr.txt` file located in the WebSpeed working directory.

4 ♦ Use the WebSpeed Configuration utility to stop the running Transaction Servers with the following command:

```
wtbman -stop ts-name
```

5 ♦ Reboot the system to make sure that the ISAPI/NSAPI messengers are released from memory.

6 ♦ Uninstall WebSpeed 3.0. Afterwards, delete any remaining Transaction Server session and error log files.

7 ♦ Install WebSpeed Version 3.1 and re-create the Transaction Servers by using the Progress Explorer. Remember that you must recompile all the 4GL files in your applications.

You can save some time re-creating your configuration by converting your previous `ubroker.properties` file to WebSpeed Version 3.1. If you rename the `ubroker.properties-sav` from your Version 3.0 installation to `ubroker.properties`, you can copy it over the Version 3.1 `ubroker.properties` file. When you start up the AdminServer, the Version 3.0 `ubroker.properties` file is automatically converted to meet WebSpeed Version 3.1 standards.

NOTE: When you install WebSpeed 3.x, either to overwrite an existing installation or to add additional Progress components to the current installation, the installation program automatically renames the existing `ubroker.properties` file, located in the `properties` directory, to `ubroker.properties-sav`. This happens on UNIX and Windows platforms.

8 ♦ Start the new WebSpeed Version 3.1 Transaction Servers. See Chapter 7, "Configuring WebSpeed On Windows," for instructions.

12.4 Coexistence Of WebSpeed Version 3.0 With WebSpeed Version 3.1

WebSpeed Version 3.0 can coexist with WebSpeed Version 3.1. However, issues will arise if migration is not performed correctly. Therefore, Progress Software recommends that you do not use mixed configurations.

Because of the way they handle memory, some systems might not let you simultaneously load two versions of the same Messenger into memory. If you must support a mixed configuration, the easiest solution is to set up a different Web server and use a different type of Messenger (CGI, ISAPI, or NSAPI) for each version. You would then have to maintain a copy of your application code for each version, changing the URLs to point to the appropriate Web server and WebSpeed Messenger.

Before you decide whether to maintain both versions, consider the following points:

- **Registry Keys**—WebSpeed Version 3.1 uses a different registry key from the key used by WebSpeed 3.0; therefore, the Version 3.0 registry key will not be overwritten.
- **Program Groups**—WebSpeed Version 3.1 uses the same default Program Group as WebSpeed 3.0. Remember to use a different Program Group name when you install WebSpeed Version 3.1.

- **Progress Explorer**—For WebSpeed Version 3.1, the Progress Explorer operates as a plug-in to the MicroSoft Management Console (MMC). The new Progress Explorer cannot connect with the AdminServer of earlier versions of WebSpeed.

- **Messengers**—The WebSpeed Version 3.1 Messengers are not intended to be backward-compatible with WebSpeed 3.0 Messengers.

  WebSpeed Active Server Pages (ASP) Messenger coexistence is not supported.

  Configurations using the NSAPI and ISAPI Messenger cannot coexist. Due to the way these Messengers work, an incoming connection cannot be correctly filtered and sent to the appropriate version of the Messenger. Therefore, only one version of these Messengers can exist per Web server. Additionally, it is not possible to configure the Netscape `obj.conf` to support two WebSpeed NSAPI Messengers.

  Because the ISAPI and CGI Messengers are integrated into the directory structure of the Web server, these two messengers are installed into the Web server `\scripts` directory. Because WebSpeed Version 3.1 Messengers use the same file name as the WebSpeed 3.0 Messengers, an installation of WebSpeed Version 3.1 overwrites the previously installed WebSpeed 3.0 Messengers.

You can work around this issue by maintaining both sets of messengers. There are two ways that you can achieve this:

– Re-name the existing WebSpeed `CGIIP.EXE` and `WSISA.DLL` messengers to a unique `.EXE` and `.DLL name`. Then install WebSpeed Version 3.1 on your system. Both sets of messengers will be in the same Scripts directory. You will then have to maintain two sets of application code, modifying the URLs to reference the new messenger name in one and leaving the other to refer to the original messenger names.

– Modify the Web server by adding an additional virtual directory to be used as a secondary Scripts directory. You can do this through your Web server administration program prior to installing WebSpeed Version 3.1 on your system. During installation, enter the new Scripts directory name when you are prompted for the Web server Scripts directory. Because the location of the messenger has changed, you need to make a copy of all your existing WebSpeed applications and modify them so that the application URLs refer to the correct Scripts directory. For example:

```
http://host/scripts/messenger/WService=broker/yourapplication
```

becomes:

```
http://host/newscripts/messenger/WService=broker/yourapplication
```

- **System Path**—The System Path presents a possible conflict. A WebSpeed 3 user might have added to the System Path a reference to the WebSpeed 3 `\bin` directory. WebSpeed 3.0 and WebSpeed Version 3.1 use some of the same command-line utilities, such as WTBMAN. The command-line utilities for Version 3.1 are not backward-compatible. If you use the system path to provide access to the WTBMAN utility, only the first WTBMAN found would be executed, and this might be the incorrect version.

  You can work around this by modifying the path specifically for a command shell. You can configure two command shells, each with a unique path and referencing the correct WebSpeed `\bin` directory. If you added the WebSpeed `\bin` directory to the system path, remove it and set it only for the specific command shell.

13

Upgrading WebSpeed On UNIX

This chapter describes the conditions and steps required to migrate from WebSpeed Version 2 or Version 3.0 to WebSpeed Version 3.1 on UNIX operating systems. The chapter includes the following sections:

- Migrating Your Existing WebSpeed Version 2 Configuration To WebSpeed Version 3.1
- Coexistence Of WebSpeed Version 2 With WebSpeed Version 3.1
- Migrating Your Existing WebSpeed Version 3.0 Configuration To WebSpeed Version 3.1
- Coexistence Of WebSpeed Version 3.0 With WebSpeed Version 3.1

NOTES: WebSpeed provides the SHOWCFG utility, which displays information on the current license. For more information, see the "SHOWCFG Utility" section in Appendix C, "Configuration and Management Utilities.".

WebSpeed also provides the Product Update utility, which lets you change the number of users or the expiration date of a Progress product once you have obtained a new License Addendum from your Progress Software sales representative. For more information, see the "Product Update Utility" section in Appendix C, "Configuration and Management Utilities.".

13.1 Migrating Your Existing WebSpeed Version 2 Configuration To WebSpeed Version 3.1

Follow these steps to migrate from WebSpeed Version 2 to WebSpeed Version 3.1:

NOTE: Follow these steps before to installing WebSpeed Version 3.1 on your system. Be sure to stop all WebSpeed applications before you uninstall WebSpeed Version 2.

1 ♦ Save the current WebSpeed Version 2 configuration. This information is stored within the `webspeed.cnf` file and can be located by looking at the WEBSPEEDCNF environment variable.

2 ♦ You should note the current WebSpeed Version 2 environment variables, specifically PROPATH. If you have customized PROPATH, write the values down and apply them to the WebSpeed Version 3.1 PROPATH. Other WebSpeed Version 2 variables might have been defined within the `wsset.env` file; its WebSpeed Version 3.1 equivalent is the `ubroker.properties` file.

3 ♦ Save any custom changes made to the Version 2 `web-disp.p` file and re-apply them to the Version 3.1 `web-disp.p` file.

4 ♦ Note any internationalization or code-page issues.

5 ♦ If you have customized your WebSpeed Messenger error messages, save the `wsCusErr.txt` file located in the WebSpeed work directory.

6 ♦ Stop each of the running Transaction Servers with the following command:

```
wtbman -stop ts-name
```

7 ♦ Delete any remaining Transaction Server session and error log files.

8 ♦ Delete the WebSpeed Version 2 installation directory.

9 ♦ Install WebSpeed Version 3.1 and modify the `ubroker.properties` file to re-create the Transaction Servers. If you have the Progress Explorer available on a remote machine, you can use it to re-create the WebSpeed Transaction Servers and NameServer. See Table 13–1 for a list of keywords that have changed from WebSpeed Version 2 to Version 3.1.

   As of WebSpeed Version 3.0, the NameServer is a mandatory component. You should review the documentation describing the NameServer. In WebSpeed Version 2, the NameServer is not required and is included only in the Enterprise version. In WebSpeed Version 2, you create the brokers. Then you have the option to create a Dispatcher, and assign brokers to the Dispatcher during configuration. In WebSpeed Version 3.1, you define the NameServer; then, as you create each Transaction Server, you assign the Transaction Server to a NameServer. Remember that you must recompile all 4GL files in your applications.

10 ♦ Start the new WebSpeed Version 3.1 Transaction Servers. See Chapter 8, "Configuring WebSpeed On UNIX," for instructions.

Keyword Conversion Chart

Table 13–1 contains a list of WebSpeed Version 2 keywords taken from the `webspeed.cnf` file and associates them with their WebSpeed Version 3.1 equivalents. You can use this chart as a guide during the process of re-creating the WebSpeed Version 2 Transaction Servers and Dispatchers as WebSpeed Version 3.1 Transaction Servers and NameServers. Progress Software recommends that you use the Progress Explorer to re-create the NameServer and WebSpeed Transaction Servers. However, you can manually edit the values used by WebSpeed Version 3.1, which are defined in the `ubroker.properties` file. It is important to note that the location of the keywords within this file is positional; moving keywords between groups could cause errors. If you choose to edit this file manually, see the "Overview Of the ubroker.properties File" section in Chapter 8, "Configuring WebSpeed On UNIX," for more information.

Table 13–1: WebSpeed Version 2 To Version 3.1 Conversion Chart For UNIX
(1 of 4)

WebSpeed Version 2 Keywords (webspeed.cnf)	WebSpeed Version 3.1 Keywords (ubroker.properties)
InstallPath—Installation directory; set by the install	Also stored in the registry; set by the install
RootPath—Web server document root directory	RootPath—Web server document root directory

Table 13–1: WebSpeed Version 2 To Version 3.1 Conversion Chart For UNIX
(2 of 4)

WebSpeed Version 2 Keywords (webspeed.cnf)	WebSpeed Version 3.1 Keywords (ubroker.properties)
ScriptPath—Web server Messenger directory	ScriptPath—Web server Messenger directory
DefaultService—Transaction Server supports the default application service	defaultService—Transaction Server supports the default application service
AllowMsngrCmds—Allow internal admin commands	AllowMsngrCmds—Allow internal admin commands
Components—Leave at default	components—Leave at default
InfoVersion—Leave at default	infoVersion—Leave at default
WsType—Values 1 for broker, 2 for dispatcher	N/A
PortNum—Transaction Server port number	portNumber—Transaction Server port number
Location—Values 200 for local, 100 for remote	location—local or remote (NameServer only)
StartUp—auto-start value	AutoStart—Yes/No
StartInstances—Initial agents to start	initialSrvrInstance—Initial Agents to start
MaxInstances—Maximum number of agents	maxSrvInstance—Maximum number of Agents
BeginNewInstances—Minimum number of free agents	minSrvrInstance—Minimum number of Agents that will be maintained
AgentMinPort—Minimum port number for an agent process	srvrMinPort—Minimum port number for an Agent
AgentMaxPort—Maximum port number for agent process	srvrMaxPort—Maximum port number for a server process
NumBrokers—Number of brokers controlled by the dispatcher	N/A

Table 13–1: WebSpeed Version 2 To Version 3.1 Conversion Chart For UNIX
(3 of 4)

WebSpeed Version 2 Keywords (webspeed.cnf)	WebSpeed Version 3.1 Keywords (ubroker.properties)
IdleTimeout—Wait period to shut down idle agents	IdleTimeout—Time Agent can be idle before Broker flags it to be trimmed if more Agents than minSvrInstance active
PriorityWeight—Transaction Server load balancing weight factor	priorityWeight—Transaction Server load balancing weight factor
HostName—System machine name	hostName—System machine name
DefaultDirectory—Application directory	workDir—Transaction Servers' working directory
ServiceExeFile—Broker executable name	ClassMain—Broker startup executable
WSISADLLFile—ISAPI messenger name	[WebSpeed.Messengers.WSISA] section; set value to msngrExecFile—to the path + executable name
WSNSADLLFile—NSAPI messenger name	[WebSpeed.Messengers.WSNSA] section; set value to msngrExecFile—to the path + executable name
CGIIPExeFile—CGI Messenger name	[WebSpeed.Messengers.CGIIP] section; set value to msngrExecFile—to the path + executable name
IISDllFIle—Renamed broker executable	N/A
AgentExeFile—Agent executable name	srvrExecFile—The agent executable name
AgentParams—Agent startup parameters	srvrStartupParam—Agent startup parameters
ErrorLogFile—Broker error log name	brokerLogFile—Broker error log name
SessionLogFile—Broker session log name	srvrLogFile—Broker session log name
User—The user ID	userName—User ID to start Transaction Server

Table 13–1: WebSpeed Version 2 To Version 3.1 Conversion Chart For UNIX
(4 of 4)

WebSpeed Version 2 Keywords (webspeed.cnf)	WebSpeed Version 3.1 Keywords (ubroker.properties)
Group—The group ID for broker process	groupName—Password to start Transaction Server
ServiceNameList—Lists brokers for dispatchers	ServiceNameList—Lists application services for the Transaction Server
Language—Not used	N/A
Environment—Production or development mode	srvrAppMode—Production or development mode
Debugging—Enabled to allow debugging	srvrDebug—Enabled to allow debugging
ApplicationURL—Default is blank	applicationURL—Default is blank
DefaultCookiePath—Default is blank	DefaultCookieDomain—Default is blank
StateTransport—Leave at default	State Transport—Leave at default
StateSavingMethod—Leave at default	StateSavingMethod—Leave at default
StateSavingDirectory—Leave at default	StateSavingDirectory—Leave at default

13.2 Coexistence Of WebSpeed Version 2 With WebSpeed Version 3.1

WebSpeed Version 2 can coexist with WebSpeed Version 3.1. However, issues will arise if you do not perform the installation correctly. Although there are fewer conflicts than in WebSpeed NT configurations, the primary problem, which is maintaining duplicate application code in order to accommodate changes to the application URL, still exists.

Because of the way they handle memory, some systems might not let you simultaneously load two versions of the same Messenger into memory. If you must support a mixed configuration, the easiest solution is to set up a different Web server and use a different type of Messenger (CGI or NSAPI) for each version. You would then have to maintain a copy of your application code for each version, changing the URLs to point to the appropriate Web server and WebSpeed Messenger.

Before you decide whether to maintain both versions, consider the following points:

- **Messengers**—WebSpeed Version 3.1 Messengers are not intended to be backward-compatible with WebSpeed Version 2 Messengers. The WebSpeed CGIIP Messenger is not copied into the Web server's Scripts directory by the install program. Instead, this messenger is executed through a shell script that is located within the Web server Scripts directory.

  Configurations using the NSAPI Messenger cannot coexist. Due to the way this Messenger works, an incoming connection cannot be correctly filtered and sent to the appropriate version of the Messenger. Therefore, only one version of this Messenger can exist per Web server. Additionally, it is not possible to configure the Netscape `obj.conf` file to support two WebSpeed NSAPI messengers.

  Both WebSpeed Version 2 and WebSpeed Version 3.1 provide a sample script called `wspd_cgi.sh` that is copied into the Scripts directory during the installation. To avoid overwriting this file, rename the original script prior to installing WebSpeed Version 3.1 on your system.

  In order to support both versions of the WebSpeed CGI Messenger, you must create an additional copy of the existing scripts, making the necessary modifications so that the environment settings are correct and the correct Messenger is executed. You must copy your application and modify the application URLs to point to the new script.

  For example:

  SYNTAX

  ```
  http://host/scripts/messenger.sh/WService=appservice-name/
  your-application
  ```

  becomes:

  SYNTAX

  ```
  http://host/scripts/newmessenger.sh/WService=appservice-name/
  your-application
  ```

- **Path**—The path presents a possible conflict. You might have added to the path a reference to the WebSpeed Version 2 `/bin` directory. WebSpeed Version 2 and WebSpeed Version 3.1 use some of the same command-line utilities, such as WTBMAN. The command-line utilities for Version 3.1 are not backward compatible. If you use the path to provide access to the WTBMAN utility, only the first WTBMAN found would be executed, and this might be the incorrect version.

  You can work around this by modifying the path specifically for a command shell. You can configure two command shells, each with a unique path and referencing the correct WebSpeed `/bin` directory. Remember, if the WebSpeed `/bin` directory was added to the system environment path, it should be removed and set only for the specific command shell.

- **Command-line Utilities**—Both WebSpeed Version 2 and WebSpeed Version 3.1 support two command-line utility programs, WTBMAN and WSCONFIG. These utilities are not backward-compatible. WebSpeed Version 3.1 also includes additional command utilities:

  – NSCONFIG is a new tool used to view the current configuration of the NameServer; it functions similarly to WSCONFIG.

  – NSMAN is a new tool used to start, stop, and query the NameServer; it functions similarly to WTBMAN.

- **Static HTML files**—Static HTML files are an issue, especially in development configurations, because both versions of the WebSpeed Workshop use static HTML files and look for them in one specific directory. Because these files have the same names, they cannot reside in the Web server's document root directory. This creates problems for coexistence.

  As a workaround, creating a second Web server on the system, with its own document root directory and running at a different port number, would solve the problem; but, again, it requires changes to the WebSpeed application URLs to incorporate the new port number.

- **Environment Variables**—When you start a Netscape Web Server, your startup script sets environment variables that are specific to each version of WebSpeed. This prevents you from simultaneously running two versions of WebSpeed on the same Netscape Web server.

13.3 Migrating Your Existing WebSpeed Version 3.0 Configuration To WebSpeed Version 3.1

Follow these steps to migrate from WebSpeed Version 3.0 to WebSpeed Version 3.1:

NOTE: Follow these steps before installing WebSpeed Version 3.1 on your system.

1 ♦ Save any custom changes made to the Version 3.0 `web-disp.p` file and re-apply them to the Version 3.1 `web-disp.p` file.

2 ♦ Note any internationalization or code-page issues.

3 ♦ If you have customized your WebSpeed Messenger error messages, save the `wsCusErr.txt` file located in the WebSpeed working directory.

4 ♦ Use the WebSpeed Configuration utility to stop the running Transaction Servers with the following command:

```
wtbman -stop ts-name
```

5 ♦ Delete any remaining Transaction Server session and error log files.

6 ♦ Delete the WebSpeed Version 3.0 installation directory.

7 ♦ Install WebSpeed Version 3.1 and modify the `ubroker.properties` file to re-create the Transaction Servers. If you have the Progress Explorer available on a remote machine, you can use it to re-create the WebSpeed Transaction Servers and NameServer. Remember that you must recompile all the 4GL files in your applications.

You can save some time re-creating your configuration by converting your previous `ubroker.properties` file to WebSpeed Version 3.1. If you rename the `ubroker.properties-sav` from your Version 3.0 installation to `ubroker.properties`, you can copy it over the Version 3.1 `ubroker.properties` file. When you start up the AdminServer, the Version 3.0 `ubroker.properties` file is automatically converted to meet WebSpeed Version 3.1 standards.

NOTE: When you install WebSpeed 3.x, either to overwrite an existing installation or to add additional Progress components to the current installation, the installation program automatically renames the existing `ubroker.properties` file, located in the `properties` directory, to `ubroker.properties-sav`. This happens on UNIX and Windows platforms.

8 ♦ Start the new WebSpeed Version 3.1 Transaction Servers. See Chapter 8, “Configuring WebSpeed On UNIX,” for instructions.

13.4 Coexistence Of WebSpeed Version 3.0 With WebSpeed Version 3.1

WebSpeed Version 3.0 can coexist with Webspeed Version 3.1. However, issues will arise if you do not perform the migration correctly. Although there are fewer conflicts than in WebSpeed NT configurations, the primary problem, which is maintaining duplicate application code in order to accommodate changes to the application URL, still exists.

If you must support a mixed configuration, the easiest solution is to set up a different Web server for each version. You would then have to maintain a copy of your application code for each version, changing the URLs to point to the appropriate Web server and WebSpeed Messenger.

Before you decide whether to maintain both versions, consider the following points:

- **Progress Explorer**—For WebSpeed Version 3.1, the Progress Explorer operates as a plug-in to the MicroSoft Management Console (MMC). The new Progress Explorer cannot connect with the AdminServer of earlier versions of WebSpeed. You can use the Progress Explorer on a remote machine to start, stop, and configure your WebSpeed components.

- **Messengers**—WebSpeed Version 3.1 Messengers are not intended to be backward-compatible with WebSpeed Version 3.0 Messengers. The WebSpeed CGIIP Messenger is not copied into the Web server's Scripts directory by the install program. Instead, this messenger is executed through a shell script that is located within the Web server Scripts directory.

  Configurations using the NSAPI Messenger cannot coexist. Due to the way this Messenger works, an incoming connection cannot be correctly filtered and sent to the appropriate version of the Messenger. Therefore, only one version of this Messenger can exist per Web server. Additionally, it is not possible to configure the Netscape `obj.conf` file to support two WebSpeed NSAPI messengers.

  Both WebSpeed Version 3.0 and WebSpeed Version 3.1 provide a sample script called `wspd_cgi.sh` that is copied into the Scripts directory during the installation. To avoid overwriting this file, rename the original script prior to installing WebSpeed Version 3.1 on your system.

In order to support both versions of the WebSpeed CGI Messenger, you must create an additional copy of the existing scripts, making the necessary modifications so that the environment settings are correct and the correct Messenger is executed. You must copy your application and modify the application URLs to point to the new script.

For example:

```
http://host/scripts/messenger.sh/WService=appservice-name/
your-application
```

becomes:

```
http://host/scripts/newmessenger.sh/WService=appservice-name/
your-application
```

- **Path**—The path presents a possible conflict. You might have added to the path a reference to the WebSpeed Version 3.0 `/bin` directory. WebSpeed Version 3.0 and WebSpeed Version 3.1 use some of the same command-line utilities, such as WTBMAN. The command-line utilities for Version 3.1 are not backward compatible. If you use the path to provide access to the WTBMAN utility, only the first WTBMAN found would be executed, and this might be the incorrect version.

  You can work around this by modifying the path specifically for a command shell. You can configure two command shells, each with a unique path and referencing the correct WebSpeed `/bin` directory. Remember, if the WebSpeed `/bin` directory was added to the system environment path, it should be removed and set only for the specific command shell.

A

Progress WebSpeed Directory Structure

The Progress installation path contains configuration files and several subdirectories. This appendix describes the subdirectories of the Progress installation path.

A.1 Installation Path Directory Structure

The installation path directory contains the Progress executables, several procedure (.p) files, and other related files and subdirectories. On Windows, the default installation path is C:\Program Files\Progress. On UNIX, the default installation path is /usr/DLC.

The installation path that you choose is referred to as DLC. 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 A–1 describes a directory tree of the Progress subdirectories.

Table A–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.
convmap	Includes files to support code-page conversions and collation tables.
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 95, Windows 98, 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.
jfc	Includes the Java Foundation Class files and executables necessary for running Progress products.
jre	Contains the Java Runtime Environment files and executables necessary for running Progress products.
lib	Contains shared objects necessary for running Progress executables.
odbc	Includes files to support ODBC.
ora	Contains files to support the DataServer for ORACLE.

Table A–1: Progress (%DLC%) Directory Structure *(2 of 2)*

Directory Name	Description
probuild	Includes files that the PROBUILD utility uses when creating custom executables.
properties	Contains property files that manage the configuration of Progress servers, such as WebSpeed, the NameServer, and the AppServer.
prohelp	Includes the online help and other necessary files for Progress.
prolang	Contains the national language support directories.
sports	Includes the schema triggers and supplier information for each sample database.
src	Contains source files for Progress ADE tools, such as the Data Dictionary, Procedure Editor, and Sample Applications.
tty	Includes mostly object (.r) files for character-mode Progress on Windows NT, Windows 95, Windows 98, and Citrix MetaFrame

.

B

Component Lists For WebSpeed Products

When you install WebSpeed products, you can choose from three installation options: Typical, Complete, or Custom. This appendix provides a list of the components and their subcomponents for the following products:

- NameServer
- NameServer Enterprise Edition
- Personal Database
- ProVision Plus
- WebSpeed Development Server
- WebSpeed Enterprise Transaction Server
- WebSpeed Messenger
- WebSpeed Transaction Server
- WebSpeed Workshop

B.1 WebSpeed 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.

B.1.1 NameServer

Table B–1 lists the NameServer components and subcomponents. Choosing the Complete or the Typical option results in the installation of all components and subcomponents listed.

Table B–1: 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
		JDK	M
		Progress Class Files	M
		ubroker tools	M

B.1.2 NameServer Enterprise Edition

Table B–2 lists the NameServer Enterprise Edition components and subcomponents. The NameServer Enterprise Edition includes load balancing, which does not come with the standard NameServer. When you choose the Complete installation option and install the NameServer Enterprise Edition, 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 B–2: NameServer Enterprise Edition Components and Subcomponents

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Common Files	M	–	–
NameServer	M	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JDK	M
		Progress Class Files	M
		ubroker tools	M

B.1.3 Personal Database

Table B–3 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.

Table B–3: Personal Database Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL	**O**	**4GL Server**	**M**
		ActiveX Control Run-time Support	**M**
		ADM Run time—CHAR	**O**
		ADM Run time—GUI	**O**
		AS/400 Client	**O**
		Base ADE	**M**
		Base Client—DA	**M**
		Character Administration Tools	**O**
		Character Client—4GL	**M**
		Character Image	**O**
		Compile Tool—CHAR	**M**
		Compile Tool—GUI	**O**
		Desktop	**M**

Table B–3: Personal Database Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL	**O**	**Graphical Administration Tools—Man**	**M**
		Graphical Client—Mandatory	**M**
		Help Compiler—GUI	**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**
Common Files	M	–	–
Legacy V8.2 DB Utilities	**O**	–	–
Legacy V8.3 DB Utilities	**O**	–	–
ProBuild Utility—Optional	**O**	–	–
ProControl Utility—Optional	**O**	–	–

Table B–3: Personal Database Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress Explorer Tools—DB	M	Administration Server	M
		Explorer Tools	M
		JRE	M
		Progress Class Files	M
Run-time Client	M	–	–
Server Networking	M	–	–
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**

B.1.4 ProVision Plus

Table B–4 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 B–4: ProVision Plus Components and Subcomponents *(1 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client—4GL	M
		Graphical Client—Mandatory	M
ActiveX Control Support	M	ActiveX Control Development Support	M
		ActiveX Control Run-time Support	M
ADE Source Code—Development	M	**4GL Profiler**	**O**
		ADE Common Source—Mandatory	M
		ADM Source Code	**O**
		DB Administration Source	**O**
		Demo Applications	**O**
		Editor Source	**O**
		ProTools Source Code	M
		Sample Application	**O**

Table B–4: ProVision Plus Components and Subcomponents *(2 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
AppBuilder—Plus	M	AppBuilder Core	M
		Base ADE	M
		Client/Server AppBuilder	M
		WebSpeed AppBuilder	M
AS/400 DataServer Client—Optional	**O**	–	–
Character Base Tools—Optional	**O**	**ADM Run time—CHAR**	**O**
		Compile Tool—CHAR—Mandatory	**M**
		Procedure Editor—CHAR	**M**
Character Database Admin Tools—Optional	**O**	–	–
Character Image—Dev—Optional	**O**	–	–
Character Run-time Client —Dev—Optional	**O**	–	–
Client Networking	**O**	–	–
Common Files	M	–	–

Table B–4: ProVision Plus Components and Subcomponents *(3 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools—Mandatory	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**
Graphical Base Tools— Dev	M	**ADM Run time—GUI**	**O**
		Base ADE	M
		Compile Tool—GUI—Mandatory	M
		Desktop	M
		Help Compiler—GUI—Mandatory	M
		Procedure Editor—GUI	M

Table B–4: ProVision Plus Components and Subcomponents *(4 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Messenger Component—Optional	**O**	–	–
NameServer—Optional	**O**	–	–
NetSetup	**O**	–	–
ODBC DataServer Client—Optional	**O**	–	–
Open Client Runtime	M	JRE	M
		Open Client Toolkit	M
		Progress Class Files	M
Open Interface Driver—Optional	**O**	–	–
Oracle DataServer Client—Optional	**O**	–	–
ProBuild Utility—Optional	**O**	–	–
ProControl Utility—Optional	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		ubroker tools	M

Table B–4: ProVision Plus Components and Subcomponents *(5 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL ESQL/C Clients—Optional	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients—Optional	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients—Optional	**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—Optional	**O**	**Report Builder**	**M**
		Report Engine	**M**
Results	**O**	**Base ADE**	**M**
		Results (Graphical)—Mandatory	**M**

Table B–4: ProVision Plus Components and Subcomponents *(6 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
SQL Database Server—Development	**O**	**Database Server**	**M**
		Database Utilities	**M**
		JDK	**M**
		Progress Class Files	**M**
		Progress Database	**M**
		SQL Server	**M**
Transaction Server—Optional	**O**	**Progress Database**	**M**
		WebSpeed Run time	**M**
		WebSpeed Server Development	**M**
Workshop Component	M	Progress Database	M
		Workshop	M

B.1.5 WebSpeed Development Server

Table B–5 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 B–5: WebSpeed Development Server Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Run-time Support	M
AS/400 DataServer Client—Optional	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools—Mandatory	M
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M
Debugger—Optional	**O**	–	–
Messenger Component	M	–	–
NameServer	M	–	–

Table B–5: WebSpeed Development Server Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ODBC DataServer Client—Optional	**O**	–	–
Open AppServer	M	Application Server	M
		Common Broker	M
		Progress Class Files	M
Oracle DataServer Client—Optional	**O**	–	–
ProBuild Utility—Optional	**O**	–	–
ProControl Utility—Optional	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		ubroker tools	M
Progress SQL ESQL/C Clients—Optional	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**

Table B–5: WebSpeed Development Server Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL JDBC Clients —Optional	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients —Optional	**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
		WebSpeed Run time	M
		WebSpeed Server Development	M

B.1.6 WebSpeed Enterprise Transaction Server

Table B–6 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 B–6: WebSpeed Enterprise Transaction Server Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client—GL	M
		Graphical Client—Mandatory	M
ADE Source Code—Optional	**O**	**4GL Profiler**	**O**
		ADE Common Source	**O**
		Demo Applications	**O**
		Editor Source	**O**
AS/400 DataServer Client—Optional	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Messenger Component	M	–	–
NameServer	M	–	–
ODBC DataServer Client—Optional	**O**	–	–

Table B–6: WebSpeed Enterprise Transaction Server Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Open AppServer	M	Application Server	M
		Common Broker	M
		Progress Class Files	M
Oracle DataServer Client—Optional	**O**	–	–
ProBuild Utility—Optional	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		ubroker tools	M
Progress SQL ESQL/C Clients —Optional	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**

Table B–6: WebSpeed Enterprise Transaction Server Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL JDBC Clients —Optional	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients —Optional	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**
Transaction Server	M	Progress Databases	M
		WebSpeed Run time	M
		WebSpeed Server Development	M

B.1.7 WebSpeed Messenger

Table B–7 lists the WebSpeed Messenger components. Choosing the Complete or the Typical option results in the installation of all components listed.

Table B–7: WebSpeed Messenger Components

Component	Mandatory/Optional
Messenger Component	M
WebSpeed Common	M

B.1.8 WebSpeed Transaction Server

Table B–8 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 B–8: WebSpeed Transaction Server Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client—GL	M
		Graphical Client—Mandatory	M
ADE Source—Optional	**O**	**4GL Profiler**	**O**
		ADE Common Source	**O**
		Demo Applications	**O**
		Editor Source	**O**
AS/400 DataServer Client —Optional	**O**	–	–
Client Networking	M	–	–

Table B–8: WebSpeed Transaction Server Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Common Files	M	–	–
Messenger Component	M	–	–
NameServer	M	–	–
ODBC DataServer Client—Optional	**O**	–	–
Open AppServer	M	Application Server	M
		Common Broker	M
		Progress Class Files	M
Oracle DataServer Client—Optional	**O**	–	–
ProBuild Utility—Optional	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		ubroker tools	M

Table B–8: WebSpeed Transaction Server Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL ESQL/C Clients —Optional	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients —Optional	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients —Optional	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**
Transaction Server	M	Progress Databases	M
		WebSpeed Run time	M
		WebSpeed Server Development	M

B.1.9 WebSpeed Workshop

Table B–9 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 B–9: WebSpeed Workshop Components and Subcomponents *(1 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client—GL	M
		Graphical Client—Mandatory	M
ActiveX Control Support	M	ActiveX Control Development Support	M
		ActiveX Control Run-time Support	M
ADE Source Code	M	**4GL Profiler**	**O**
		ADE Common Source —Mandatory	M
		DB Administration Source	**O**
		Demo Applications	**O**
		Editor Source	**O**
		ProTools Source Code	M

Table B–9: WebSpeed Workshop Components and Subcomponents *(2 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
AppBuilder—WebSpeed	M	AppBuilder Core	M
		Base ADE	M
		WebSpeed AppBuilder	M
AS/400 DataServer Client —Optional	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools—Mandatory	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**
Messenger Component—Optional	**O**	–	–

Table B–9: WebSpeed Workshop Components and Subcomponents *(3 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
NameServer—Optional	**O**	–	–
NetSetup	**O**	–	–
ODBC DataServer Client—Optional	**O**	–	–
Oracle DataServer Client—Optional	**O**	–	–
ProBuild Utility—Optional	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		ubroker tools	M
Progress SQL ESQL/C Clients —Optional	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**

Table B–9: WebSpeed Workshop Components and Subcomponents *(4 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL JDBC Clients—Optional	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients—Optional	**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—Optional	**O**	**Progress Databases**	**M**
		WebSpeed Run time	**M**
		WebSpeed Server Development	**M**

Table B–9: WebSpeed Workshop Components and Subcomponents *(5 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
WebSpeed Workshop	M	Procedure Editor—GUI	M
		Progress Databases	M
		Workshop	M

C

Configuration and Management Utilities

This appendix discusses the purpose and use of the configuration and management utilities in the following sections:

- NSCONFIG Utility
- NSMAN Utility
- WSCONFIG Utility
- WTBMAN Utility
- WebSpeed Error Customization Utility
- SHOWCFG Utility
- Product Update Utility

Figure C–1 shows the conventions used in utility command syntax.

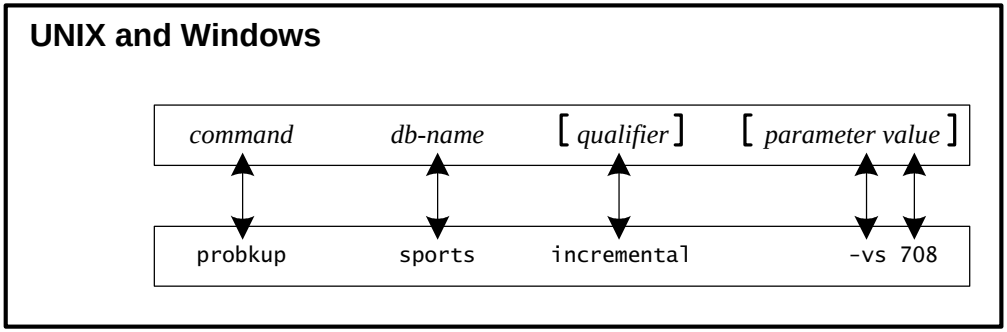


Figure C–1: Utility Syntax Conventions

Table C–1 describes each of the command components.

Table C–1: Progress Command Components

Component	Description
command	Progress executable
qualifier	Additional command specification
parameter	Operating criteria for the command
value	Numeric value or file specification

NOTE: Enter parameters on UNIX and Windows exactly as shown in the syntax descriptions.

C.1 NSCONFIG Utility

Use the NSCONFIG utility to help validate existing NameServer configurations defined in a properties file, such as the `ubroker.properties` file. This utility displays the property settings associated with a NameServer configuration, and checks that the syntax and values are valid.

The NSCONFIG utility runs locally, on the machine where the NameServer is running. The utility does not run across the network.

SYNTAX

Operating System	Syntax
UNIX Windows	`nsconfig [[[-name name-server] [-propfile path-to-properties-file] [-validate]]	-help]`

PARAMETERS

`-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 Windows 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.

EXAMPLES

Table C–2 shows several examples that use the NSCONFIG command. Assume the NameServer is `NS1`.

Table C–2: NSCONFIG Command Examples

Task	Commands
View a NameServer configuration.	`nsconfig -name NS1`
View a NameServer configuration defined within a different property file.	`nsconfig -name NS1 -propfile g:\other.properties -validate`

NOTE

A single NameServer can simultaneously support all of the AppServer, WebSpeed, and DataServer products using Progress Explorer.

C.2 NSMAN Utility

Use the NSMAN utility to control the operation of a configured NameServer. The utility allows you to start a NameServer, query its status, and shut down a NameServer.

SYNTAX

Operating System	Syntax

UNIX
Windows
	nsman
	{
	{ -name *name-server*
	{
	-kill
	| -start
	| -stop
	| -query
	}
	[
	-host *host-name* -user *user-name*
	| -user *user-name*
	]
	[-port *port-number*]
	}
	| -help
	}

PARAMETERS

`-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 username and password.

`-port` *`port-number`*

Specifies the port number of the machine where the AdminServer is running. If a port number is not specified, it defaults to 20931.

`-help`

Displays command-line help.

NOTE

Table C–3 shows several examples that use the NSMAN command. Assume the NameServer is `NS1`; the user name is `tom`; and the AdminServer is on the remote host `finance` on the port `9999`.

Table C–3: NSMAN Command Examples *(1 of 2)*

Task	Commands
Start a local NameServer.	`nsman -name NS1 -start`
Start a remote NameServer.[1]	`nsman -name NS1 -host finance -port 9999 -user tom -start`
Query a local NameServer.	`nsman -name NS1 -query`
Query a remote NameServer.[1]	`nsman -name NS1 -host finance -port 9999 -user tom -query`

Table C–3: NSMAN Command Examples *(2 of 2)*

Task	Commands
Stop a local NameServer.	`nsman -name NS1 -stop`
Stop a remote NameServer.[1]	`nsman -name NS1 -host finance -port 9999 -user tom -stop`

[1] Prompts for a password.

NOTES

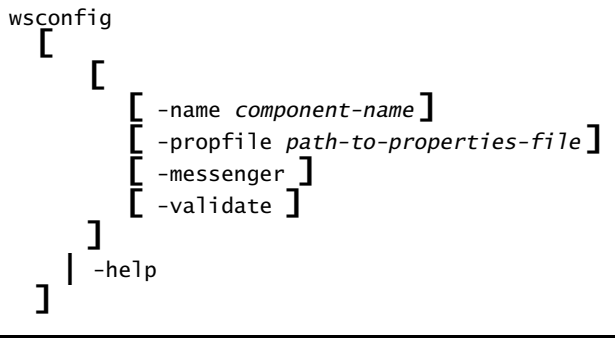
- A single NameServer can simultaneously support all of the AppServer, WebSpeed, and DataServer products.
- When you specify a user name with the `-user` parameter, Windows NT supports three different formats:
  - A user name as a simple text string, such as "`mary`," implies a local user whose user 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.

C.3 WSCONFIG Utility

Use the WSCONFIG utility to help validate existing WebSpeed Transaction Server or WebSpeed Messenger configurations defined in a properties file, such as the `ubroker.properties` file. This utility displays the property settings associated with a WebSpeed Transaction Server or Messenger configuration, and checks that the syntax and values are valid.

The WSCONFIG utility runs locally, on the machine where the WebSpeed components that you want to check are running. Because the utility does not run across the network and no AdminServer is installed during a Messenger-only install, you cannot use the WSCONFIG utility to check a Messenger-only install. The WSCONFIG utility uses the following syntax:

SYNTAX

Operating System	Syntax
UNIX **Windows**	

PARAMETERS

`-name` *component-name*

Specifies the name of an existing WebSpeed Transaction Server or Messenger configuration to examine. The name must match a name of an existing WebSpeed Transaction Server configuration defined in the specified properties file. Although you must specify a Transaction Server, you need not specify a Messenger. If you do not specify any name, the WSCONFIG utility analyzes all the WebSpeed Transaction Server and Messenger 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 Windows NT
- `$DLC/properties/ubroker.properties` on UNIX.

`-messenger`

Displays one or all of the Messengers for you to view. If `-name` refers to a Messenger and the `-messenger` parameter is used, then information appears for that **one** Messenger. If `-name` does not refer to a Messenger and the `-messenger` parameter is used, then information appears for **all** the Messengers.

The Messenger names on Windows NT are CGIIP, WSISA, WSNSA, and WSASP. The Messenger names on UNIX are CGIIP and WSNSA.

`-validate`

Checks the syntax and values of property settings defined in the specified properties file.

`-help`

Displays command-line help.

EXAMPLES

Table C–4 shows several examples that use the WSCONFIG command. Assume the Transaction Server name is `wsbroker1`.

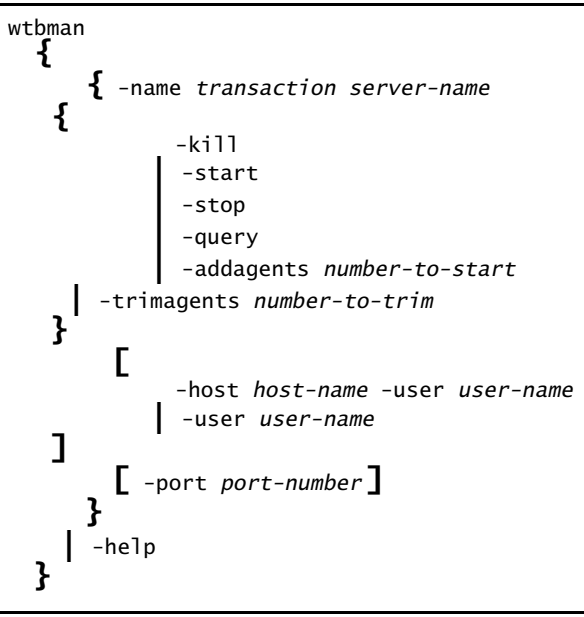
Table C–4: WSCONFIG Command Examples

Task	Commands
View a Transaction Server configuration.	`wsconfig -name` *`wsbroker1`*
View a messenger configuration.	`wsconfig -name` *`CGIIP`* `-messenger`
View all messenger configurations.	`wsconfig -messenger`
Validate the syntax and view the configuration of all messengers defined within a different property file.	`wsconfig -propfile` *`g:\other.properties`* `-validate`

C.4 WTBMAN Utility

Use the WTBMAN utility to control the operation of a configured WebSpeed Transaction Server. The utility allows you to start a Transaction Server, query its status, start and stop additional WebSpeed Agents, trim by a certain number of agents, and shut down the Transaction Server.

SYNTAX

Operating System	Syntax
UNIX Windows	

PARAMETERS

`-name` *transaction server-name*

Specifies the name of a Transaction Server.

`-kill`

Stops and removes the NameServer from memory, no matter what it is doing.

`-start`

Starts the Transaction Server.

`-stop`

Tells the NameServer to stop itself.

`-query`

Queries the Transaction Server for its status.

`-addagents` *`number-to-start`*

Specifies the number of additional agents to start.

`-trimagents` *`number-to-trim`*

Specifies the number of additional agents to trim.

`-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 when logging into a remote machine. 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.

`-port` *`port-number`*

Specifies the port number of the machine on which the AdminServer controlling the WebSpeed Transaction Server is running. If a port number is not specified, it defaults to 20931.

`-help`

Displays command-line help.

EXAMPLES

Table C–5 shows several examples that use the `wtbman` command. Assume that the Transaction Server name is `wsbroker1`; the user name is `tom`; and the AdminServer is on the remote host `finance` at port `9999`.

Table C–5: WTBMAN Command Examples

Task	Commands
Start a local Transaction Server.	`wtbman -name wsbroker1 -start`
Start a remote Transaction Server.[1]	`wtbman -name wsbroker1 -host finance -port 9999 -user tom -start`
Query a local Transaction Server.	`wtbman -name wsbroker1 -query`
Query a remote Transaction Server.[1]	`wtbman -name wsbroker1 -host finance -port 9999 -user tom -query`
Add agents (for example, 2) to a local Transaction Server.	`wtbman -name wsbroker1 -addagents 2`
Add agents (for example, 2) to a remote Transaction Server.[1]	`wtbman -name wsbroker1 -host finance -port 9999 -user tom -addagents 2`
Trim agents (for example, 3) from a local Transaction Server.	`wtbman -name wsbroker1 -trimagents 3`
Trim agents (for example, 3) from a remote Transaction Server.[1]	`wtbman -name wsbroker1 -host finance -port 9999 -user tom -trimagents 3`
Stop a local Transaction Server.	`wtbman -name wsbroker1 -stop`
Stop a remote Transaction Server.[1]	`wtbman -name wsbroker1 -host finance -port 9999 -user tom -stop`

[1] Prompts for a password.

NOTE

When you specify a user name with the `-user` parameter, Windows NT supports three different formats:

- 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.

C.5 WebSpeed Error Customization Utility

The WebSpeed Error Customization Utility allows you to change any Progress error message that is passed to the WebSpeed Messenger.

In general, the WebSpeed Messenger passes default error messages to client browsers when WebSpeed requests generate errors. Using the WebSpeed Error Customization Utility, you can:

- Retain the default error message
- Replace the default error message with a message that you create
- Pass a URL to the client browser, which directs the client browser to a specified web site

To access the WebSpeed Error Customization Utility, you must enable the Administration Utility. You can do this through the Progress Explorer by choosing Internal administration command-WSMAdmin in the Messenger Properties dialog box. Or, you can add the following line to the `[WebSpeed.Messengers.`*`messenger type`*`]` section of the `ubroker.properties` file:

```
AllowMsngrCmds=1
```

After you have enabled the Administration Utility, use a Web browser to go to the WebSpeed Messenger Administration Page on any machine that has access to the utility. Use a URL with the following format:

```
http://host_name[:port]/scripts_dir/messenger/WService=broker?WSMAdmin
```

When you have accessed the WebSpeed Messenger Administration Page, select Customize Messenger Error Messages, which displays the WebSpeed Error Customization Utility shown in Figure C–2.

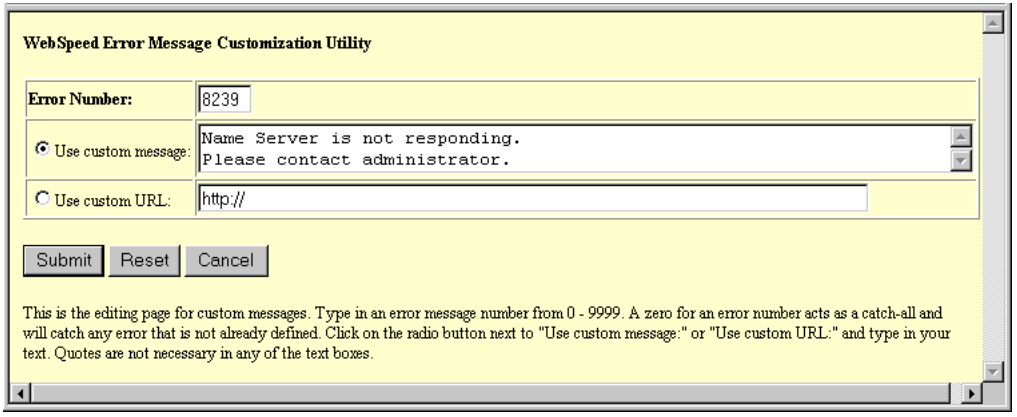


Figure C–2: WebSpeed Error Message Customization Utility

Within the customization utility, you can either accept defaults, create your own error messages, or redirect client browsers to a specified URL. Any changes you make are written to the Messenger's `wsCusErr.txt` file which is created in the Messenger's log file directory. You do not need to restart the Messenger to make your changes effective.

NOTE: There is a limitation on how many error messages you can customize. Currently, the `wsCusErr.txt` file can contain customizations for no more than 32 error messages.

Another way to customize error messages is to directly edit the `wsCusErr.txt` file, which, if it does not already exist, you can create in the Messenger's log file directory. Use any editor that allows you to save as an unformatted text file.

Entries in the wsCusErr.txt file have the following syntax:

```
error_number 0|1|2
message|URL
```

error_number

A number that represents a Progress error. If *error_number* is 0, it will match any Progress error number that is not specified in the file. You can use an entry with an *error_number* of 0 to create a default message for any errors that you did not explicitly customize.

0|1|2

0 causes the default Progress error message to be displayed. 1 causes the text string specified in the next line to be displayed. 2 causes the URL specified in the next line to be passed to the client browser.

message|*URL*

message is a text string representing the customized error message that you want to send to the browser. *URL* is a text string representing the URL for a valid web site.

The following is an example of the contents of a wsCusErr.txt file:

```
8239 1
Name Server is not responding. Please contact administrator.

0 2
http://www.progress.com
```

The first entry causes the specified messages to be displayed whenever an 8239 error is generated. The second entry directs the client browser to the specified web site whenever any other error is generated.

C.6 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.

The following sections describe how to use the SHOWCFG utility on Windows and on UNIX.

C.6.1 Using the SHOWCFG Utility On Windows

On Windows, 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 C–3 shows a typical display of the **Product Configuration Information** dialog box.

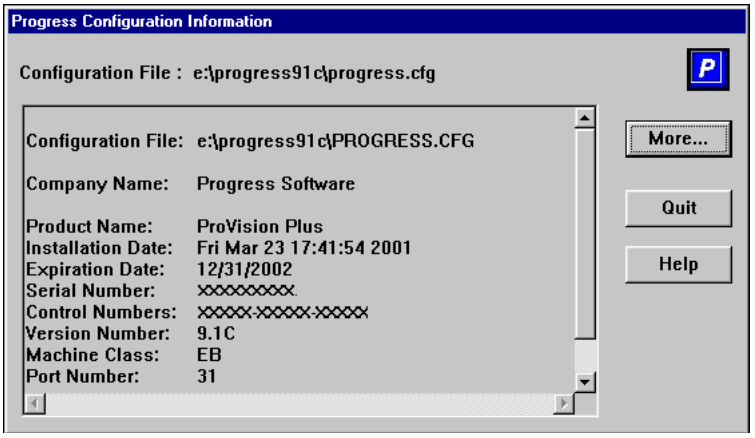


Figure C–3: Product Configuration Information From the Windows Version Of the SHOWCFG Utility

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 configuration (`.cfg`) file other than `DLC:PROGRESS.CFG`.

C.6.2 Using the SHOWCFG Utility On UNIX

To start the SHOWCFG utility on UNIX, use the following syntax at the command line:

SYNTAX

```
showcfg $env_variable/progress.cfg
```

For example:

```
showcfg $DLC/progress.cfg
```

The SHOWCFG utility displays the product configuration information stored in the Progress Version 9.1 configuration file `progress.cfg`, which is created and modified during product installation.

Figure C–4 shows a typical display of the product configuration.

```
Configuration File:  progress.cfg

      Company Name:  Progress

      Product Name:  4GL Development
 Installation Date:  Thu Apr  5 15:35:48 2001
   Expiration Date:  12/31/2002
     Serial Number:  123456789
   Control Numbers:  9VYG9 - 7AQ7N - BVG7J
    Version Number:  9.1C
     Machine Class:  EB
       Port Number:  11
```

Figure C–4: Product Configuration Displayed Using the SHOWCFG Utility On UNIX

DISPLAY FIELDS

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.

C.7 Product Update Utility

The Product Update utilitylets you change the number of users or the expiration date of a Progress product once you have obtained a new License Addendum from your Progress Software sales representative. The following sections describe how to use the Product Update utility on Windows and on UNIX.

NOTE: You cannot use the Product Update utility to add additional Progress products.

C.7.1 Using the Product Update Utility On Windows

To user the Product Update utility on Windows, follow these steps:

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.

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**.

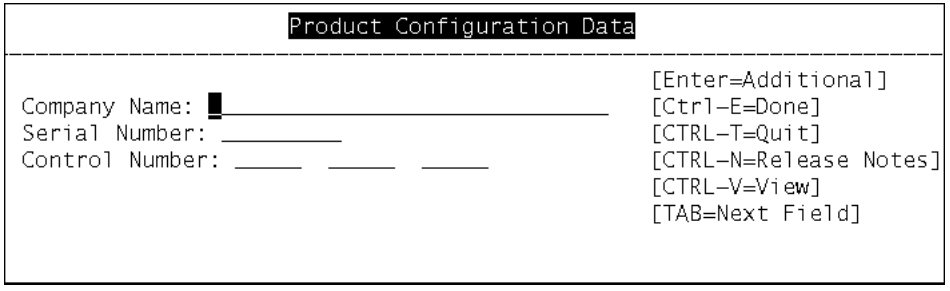
C.7.2 Using the Product Update Utility On UNIX

To use the Product Update utility on UNIX, follow these steps:

1 ♦ Change your current working directory to the directory where you installed Progress.

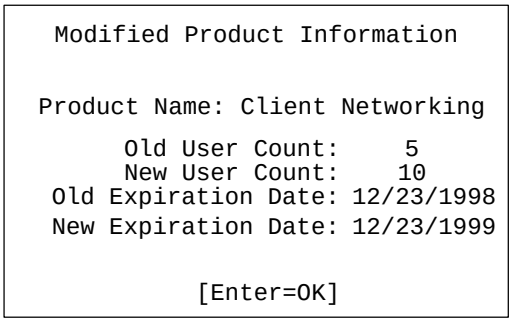
2 ♦ At the system prompt, type **proupdt** and press **RETURN**.

3 ♦ When a welcome message appears, press **RETURN**. The **Product Configuration Data** dialog box appears:



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

5 ♦ Press **ENTER**. The **Modified Product Information** dialog box appears:



Press **ENTER** again to return to the **Product Configuration Data** dialog box.

6 ♦ Press **CTRL-E** to indicate that you are done.

NOTE: You cannot press **CTRL-E** from the **Serial Number** field.

Index

A

Administration framework 7–4, 8–4

AdminServer
  defined 2–5, 7–4, 8–5
  starting on UNIX 8–18
  starting on Windows 7–13
  starting with PROADSV utility 2–5, 8–26

Agent
  AVAILABLE status 7–22, 8–28
  BUSY status 7–22, 8–28
  code page 10–13
  defined 2–2
  error messages 10–13
  international configurations 10–12
  language 10–13
  LIMBO status 7–22, 8–28
  LOCKED status 7–22, 8–28
  starting dynamically 7–24, 8–29
  STARTING status 7–22, 8–28
  trimming running 7–24, 8–29

AppBuilder
  types defined 2–3

Architecture
  WebSpeed 2–1 to 2–9

Audience i–xiii

Audio files (.au)
  stored on Web server 7–12, 8–15

B

Batch mode
  setup.iss 5–30

Bold typeface
  as typographical convention i–xvi

Broker
  defined 2–2

C

CGI Messenger
  configuring on UNIX 8–12
  executable 7–9, 8–8
  invoking on UNIX 8–13
  V2 and V3.1 coexistence on UNIX 13–7
  V2 and V3.1 coexistence on Windows 12–7
  V3 and V3.1 coexistence on UNIX 13–11
  V3 and V3.1 coexistence on Windows 12–11

Character sets
  international settings 5–11, 5–21

Code pages 10–13

Coexistence
  WebSpeed V2 and V3.1 on UNIX 13–6
  WebSpeed V2 and V3.1 on Windows 12–6
  WebSpeed V3 and V3.1 on UNIX 13–11
  WebSpeed V3 and V3.1 on Windows 12–10

Command line
  setup.iss
    batch mode 5–30

Command-line utilities
  for WebSpeed 2–6
  NSCONFIG 7–8, 8–7
  NSMAN 7–7, 8–6
  WSCONFIG 7–7, 8–6
  WTBMAN 7–6, 8–5

Commands
  mount 6–2
  proinst 6–4
  unmount 6–20, 6–30

Compatibility
  WebSpeed Agent and Database Server 1–10

Complete installation
  described 6–10
  on UNIX 6–10 to 6–20
  on Windows 5–8 to 5–14

Components
  listed by product B–1

Configurations
  deployment network with separate Web server 3–14
  development network with central machine 3–7
  development network with separate Web server 3–9
  Messenger-only installs 7–20, 8–25
  multiple LAN development environment 3–10
  restarting 11–2
  shared development/deployment network 3–16
  single-machine 3–5
  UNIX overview 8–3
  validating with status.p 7–23, 8–28
  Windows overview 7–3

CONNECT statement 10–3

Connection limits
  Agents to Transaction Servers 4–6
  for Progress databases 4–5

Control number
  displaying C–18

Control number, displaying C–20

Custom installation
  described 5–15, 6–10
  selecting components on UNIX 6–23
  selecting components on Windows 5–16
  selecting products on UNIX 6–21
  selecting subcomponents on Windows 5–17

D

Data integrity
  restarting configurations 11–2

Database
  code page 10–13
  connecting 10–2
  connecting locally 10–3
  connecting programmatically 10–3
  connecting remotely 10–4
  connection limits 4–5
  multi-national 10–9
  sample application A–3
  shared memory connections 10–3
  Sports2000 10–2

Database server
  starting 10–2

DataServers
and WebSpeed architecture 10–6
connecting with 10–4
local 10–6
managing Agent connections 10–8
remote 10–7

Date (-d) startup parameter 10–13

Date format
international settings 5–11, 5–21

DB2 database
connecting to 10–4

DB2/400 database
connecting to 10–4

Destination pathname
options 6–11, 6–12

Directory structure
WebSpeed A–1 to A–3

Disk space
Custom install on UNIX 6–21
insufficient space warning 6–17, 6–28
remaining after UNIX install 6–21

Disk space requirements
on UNIX 1–9
on Windows 1–8

DLC
directory structure A–2

DLC environment variable
setting for NSAPI Web server 8–11

Document root directory 5–5, 6–7
defined 4–3
specifying 5–9
static HTML files in 7–12, 8–14

Dynamic code-page support 9–1

E

Environment variables
PROMSGS 10–11
PROPATH 7–14, 8–19
V2 and V3.1 coexistence on UNIX 13–8
WRKDIR 7–14, 8–19

Error messages
displaying descriptions i–xxi

European Numeric Format (-E) startup parameter 10–13

Expiration date
displaying C–18

Expiration date, displaying C–20

F

File descriptors
UNIX requirements for WebSpeed 1–9

G

GENUUID utility 8–17

H

Hardware requirements 1–3

Help
WebSpeed messages i–xxi

Host name
defined 4–2

Host Name (-H) startup parameter
database connections 10–3
remote database connections 10–4

HTML files
location 5–9, 6–14, 6–25

I

IIS
virtual document root directory 5–9

Image files (.gif)
stored on Web server 7–12, 8–15

Informix database
connecting to 10–4

initialization file 6–31

Initialization File (-ininame) startup parameter 10–13

Installation
adding products on Windows 5–29
before you begin 4–7
Complete installs on UNIX 6–10 to 6–20
Complete installs on Windows 5–8 to 5–14
component lists B–1
date
displaying C–18
date, displaying C–20
levels 3–3
planning 4–5
preliminary steps on UNIX 6–2 to 6–9
preliminary steps on Windows 5–2 to 5–7
specifying Web server type on UNIX 6–13
steps for Custom on UNIX 6–20 to 6–30
steps for Custom on Windows 5–15 to 5–28
Typical installs on UNIX 6–10 to 6–20
Typical installs on Windows 5–8 to 5–14

installation
initialization file 6–31

Installation path (DLC)
directory structure A–2

Internal Code Page (-cpinternal) startup parameter 10–13

International settings
choosing 5–11, 5–21, 6–15

ISAPI Messenger
executable 7–9
V2 and V3.1 coexistence on Windows 12–7
V3 and V3.1 coexistence on Windows 12–11

ISS batch mode 5–30 to 5–31

Italic typeface
as typographical convention i–xvi

ITP applications

J

Java
  class files location 7–12, 8–14
  JavaHome Pathname 6–8

Java requirements
  in Japanese environments 1–5

JavaHome Pathname 6–8

JDK/JRE requirements 1–4

K

Keystrokes i–xvi

Keywords
  conversion on UNIX 13–3
  conversion on Windows 12–3

L

Language
  selecting default 5–10, 5–20, 6–15, 6–25

License
  updating C–21
  updating information C–21

Licensing 5–2

Licensing information
  displaying 1–10

Load balancing
  Enterprise NameServer 2–4
  NameServer 7–16, 8–21

Log files
  configuring 7–15, 8–20
  maintaining 7–20, 8–25

M

Mandatory components
  in Custom installs 5–17, 6–23

Manual
  organization of i–xiii
  syntax notation i–xvii

Messages
  displaying descriptions i–xxi

Messenger
  components list B–19
  defined 2–2
  downloading executables 7–9, 8–8
  executable location on UNIX 8–8
  executable location on Windows 7–8
  executables 4–6
  managing 7–25, 8–30
  operating system compatibility 7–9, 8–8
  script file location 7–12, 8–14
  V2 and V3.1 coexistence on UNIX 13–7
  V2 and V3.1 coexistence on Windows 12–7
  V3 and V3.1 coexistence on UNIX 13–11
  V3 and V3.1 coexistence on Windows 12–11

Microsoft IIS
  Messenger executable 7–9

Microsoft Management Console
  Progress Explorer plugin 2–6

Migration
  keyword conversion on UNIX 13–3
  keyword conversion on Windows 12–3
  WebSpeed V2 to V3.1 on UNIX 13–2
  WebSpeed V2 to V3.1 on Windows 12–2
  WebSpeed V3 to V3.1 on UNIX 13–9
  WebSpeed V3 to V3.1 on Windows 12–9

Monospaced typeface
  as typographical convention i–xvi

Mount commands
for UNIX install 6–2

Multi-user database 10–2

N

NameServer
component list B–2
defined 2–3
linking with other NameServers 3–10
load balancing 7–16, 8–21
predefined instance 7–6
replaces Dispatcher 12–3, 13–3
starting 7–18, 8–23
validating configurations 8–18
Windows system requirements 1–7

NameServer Enterprise Edition
components list B–3

Netscape
environment variables coexistence issue 13–8
Messenger executable 7–9, 8–8
multiple Messengers 12–7

Networking requirements 1–3

NSAPI Messenger
executable 7–9, 8–8
settings on UNIX 8–11
V2 and V3.1 coexistence on UNIX 13–7
V2 and V3.1 coexistence on Windows 12–7
V3 and V3.1 coexistence on UNIX 13–11
V3 and V3.1 coexistence on Windows 12–11

NSAPI-type Web server
configuring WebSpeed on 7–10, 8–9
obj.conf file 7–10, 8–10
start file 8–11

NSCONFIG utility
described 7–8, 8–7
syntax and notes C–3

NSMAN utility
described 7–7, 8–6
syntax and notes C–5

n-tier deployment
Messenger-only installs 7–20, 8–25

Number format
international settings 5–11, 5–21

O

obj.conf file
modifying 7–10, 8–9
sample 7–11, 8–10

ORACLE database
connecting to 10–4

P

Parameter file
starting database with 10–2

Path
V2 and V3.1 coexistence on UNIX 13–8, 13–12

Personal Database
components list B–4

ping command
testing network connections 4–4

Preinstallation tasks 4–7

priorityWeight property 7–16, 8–21

PRO*Tools 2–3

PROADSV utility
starting AdminServer 8–5, 8–26

PROCFG environment variable
setting for NSAPI Web server 8–11

Product Configuration Data 6–4

Product Life Cycle 1–12
  Active 1–12
  Functionally Stable 1–12
  Mature 1–12
  Retired 1–13

Product name
  displaying C–18, C–20

Product Update utility C–21

Progress Explorer
  defined 2–6
  described 7–5, 8–5
  using to edit configuration 7–13, 8–15
  using with DataServers 10–7
  V3 and V3.1 coexistence 12–11, 13–11
  validating configurations 7–21, 8–27

progress.cfg file
  displaying 1–10

PROMSGS environment variable
  international configurations 10–11
  limits on NSAPI Web server 10–11
  removing prior to a reinstall 4–7
  setting for NSAPI Web server 8–11

PROPATH environment variable 7–14, 8–19

Property file
  overview 7–13, 8–15

ProVision Plus 3–2
  components list B–7

ProVision Plus AppBuilder
  defined 2–3

ProVision Plus Development Server
  Windows system requirements 1–7

R

Registry keys
  V2 and V3.1 coexistence on Windows 12–7
  V3 and V3.1 coexistence on Windows 12–10

Response file
  creating automatically 5–30
  creating manually 5–30

S

Sample application database A–3

Schema holder 10–7

Script directory
  copying to bin directory 6–18
  defined 4–3
  specifying 5–9

Script file
  wspd_cgi.sh shell 8–12

Serial number
  displaying C–18

Serial number, displaying C–20

Service Name (-S) startup parameter
  database connections 10–3
  remote database connections 10–4

Shared memory
  database connection 10–3

Show Configuration (SHOWCFG) utility C–21

SHOWCFG utility 1–10, C–17

Single-user database 10–2

Sports2000 database 10–2

Starting configurations
  deployment network with separate Web server 11–8
  development network with central machine 11–4
  development network with separate Web server 11–5
  multiple LAN development environment 11–6
  shared development/deployment network 11–9
  single-machine 11–3

Static files
  coexistence issues on UNIX 13–8
  coexistence issues on Windows 12–9
  HTML 7–12, 8–14
  Java class files 7–12, 8–14
  location 5–9, 6–14, 6–25

Status.p procedure
  validating configurations 7–23, 8–28

Stream Code Page (-cpstream) startup parameter 10–13

Syntax notation i–xvii

System path
  V2 and V3.1 coexistence on Windows 12–8, 12–12

System requirements 1–2 to 1–11
  hardware requirements 1–3
  Java 1–4
  networking requirements 1–3

T

Tools
  PRO*Tools 2–3
  WebTools 2–4

Type, Device, and Destination dialog box 6–9

Typical installation
  described 6–10
  on UNIX 6–10 to 6–20
  on Windows 5–8 to 5–14

Typographical conventions i–xv, i–xvi

U

ubroker.properties file
  editing 8–16
  hierarchy 8–15
  location 2–7, 8–16
  overview 2–7, 7–13, 8–15
  structure 8–15
  unique values needed 8–16
  validating edits 8–18

ULIMIT
  setting before installation 4–7
  WebSpeed Agents requirement 1–9

Uninstalling WebSpeed
  on Windows 5–34

Unmount command
  for UNIX install 6–20, 6–30

Updating
  license information C–21
  number of users C–21

Upgrades
on Windows installations 5–34

URL
international Web site 10–14

uuid
in ubroker.properties 8–17

V

Validating
WebSpeed configurations 7–21, 8–26

Version number
displaying C–18

Version number, displaying C–20

Video files (.mpeg, .mpg)
stored on Web server 7–12, 8–15

Virtual directory
IIS 5–9

W

Web browser
requirements 1–3

Web server
and Messenger types 4–6
document Root directory 5–5
document root directory 4–3, 6–7
home page 4–4
Host Name 4–2
NSAPI type 7–10, 8–9
overview 4–2
requirements 1–3
restarting 4–3
script directory 4–3, 5–5
scripts directory 6–7, 6–13
setting up 7–8, 8–7
testing network connections 4–4
type 4–2, 5–5, 6–7

Web site
international 10–9
multi-lingual 10–9

WebSpeed
architecture 2–1 to 2–9
configuration on UNIX 8–3
configuration on Windows 7–3
directory structure A–1 to A–3
licensing information 1–10
planning installations 4–5
running on UNIX 8–26
running on Windows 7–21
testing after configuration 7–21, 8–26
Web request flow chart 2–8

WebSpeed administration framework 7–4, 8–4

WebSpeed Agent
defined 2–2

WebSpeed AppBuilder
defined 2–3
Windows system requirements 1–6

WebSpeed architecture
requirements 2–9

WebSpeed Broker
defined 2–2
starting 8–24

WebSpeed components
overview 2–2
UNIX system requirements 1–9
Windows system requirements 1–6

WebSpeed Development Server
components list B–13

WebSpeed Dispatcher
replaced by NameServer 12–3, 13–3

WebSpeed Enterprise Transaction Server
components list B–16

WebSpeed Messenger
defined 2–2
Windows system requirements 1–6

WebSpeed Transaction Server
Agent connection limits 4–6
components list B–19
Development Server defined 2–4
Enterprise Transaction Server defined 2–4
forcing immediate shut down 7–24, 8–30
managing 7–23, 8–29
sample instance 7–6
starting 7–19
stopping with WTBMAN 7–24, 8–30
Transaction Server defined 2–4
validating configurations 8–18
Windows system requirements 1–7

WebSpeed WebTools
installing on separate Web server 3–10

WebSpeed Workshop 3–2
components list B–22

WebSpeed Workshop Environment
defined 2–3

WebTools 2–4

Weight factor
load balancing 7–16, 8–21
setting to zero 7–17, 8–22

Weight factors
arbitrary 7–17, 8–22
fail-over 7–17, 8–22
percentage-based 7–16, 8–21

Working directory 5–6
environment variable 7–14, 8–19

WRKDIR environment variable
setting for NSAPI Web server 8–11

WSASP Messenger
executable 7–9
V2 and V3.1 coexistence on Windows 12–7
V3 and V3.1 coexistence on Windows 12–11

WSCONFIG utility
described 7–7, 8–6
syntax and notes C–8

WSMAdmin page
URL 7–25, 8–30
validating configurations 7–4, 8–4
viewing 7–23, 8–28

wspd_cgi.sh script
V2 and V3.1 coexistence on UNIX 13–7

WTBMAN utility
described 7–6, 8–5
syntax and notes C–10