

WebSpeed
Product Update Bulletin

© 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: 4415
Item Number: 89441;V3.1D

Contents

Preface ix
Purpose ix
Audience ix
Organization Of This Manual ix
How To Use This Manual x
Typographical Conventions xi
Other Useful Documentation xii
Getting Started xii
Development Tools xiii
Reporting Tools xv
4GL xv
Database xvi
DataServers xvii
SQL-89/Open Access xvii
SQL-92 xviii
Deployment xix
WebSpeed xix
Reference xx

1. Overview Of New and Revised Features 1–1
1.1 Changes In Version 3.1D 1–2
1.1.1 Dynamic Code Page Support 1–2
1.1.2 AdminServer Security Enhancement 1–3
1.1.3 Optional NameServer Enhancement 1–4
1.1.4 New Environment Variables For the WebSpeed Broker 1–5
1.1.5 New WebTools 1–6
1.1.6 Enhanced Diagnostic Logging For WebSpeed Messenger . . . 1–7
1.1.7 The ping.p, reset.p, and debug.p Utilities Moved 1–7

1.2 Changes In Version 3.1C 1–8
1.2.1 Java Requirements 1–8
1.2.2 Enhancements To the WebSpeed Error Customization Utility 1–8
1.3 Changes In Version 3.1B 1–9
1.3.1 Saving Changes To the ubroker.properties File 1–9
1.3.2 Customizing Error Messages 1–9
1.3.3 OS Locale Startup Parameter 1–9
1.3.4 Development Tools 1–10
1.3.5 SpeedScript 1–12
1.3.6 Virtual Directories 1–12
1.4 Changes In Version 3.1A 1–13
1.4.1 Installation 1–13
1.4.2 Administration Tools 1–14
1.4.3 Development Tools 1–17
1.4.4 SpeedScript 1–22
1.4.5 Internationalization 1–26
1.4.6 XML 1–30
1.4.7 Sockets 1–32

2. Documentation 2–1
2.1 Electronic Documentation (EDOC) 2–2
2.1.1 DynaText 2–2
2.1.2 PDF 2–2
2.1.3 Dynamic HTML 2–2
2.2 Summary Of Version 3.1D Documentation Changes 2–3
2.3 Summary Of Version 3.1C Documentation Changes 2–5
2.4 Summary Of Version 3.1B Documentation Changes 2–5
2.5 Summary Of Version 3.1A Documentation Changes 2–5

A. WebSpeed Security Issues A–1
A.1 Changing WebSpeed Applications From Development Mode To Production Mode A–2
A.1.1 Set the Agent Application Mode To Production A–2
A.1.2 Disable the WebSpeed Messenger Administration Utility A–5
A.1.3 Establish the WebSpeed Messenger Administration Internet Protocol List (Optional) A–8
A.1.4 Check the Status Of the Debug Mode A–11
A.2 Changing Additional Settings To Minimize Security Risks A–14
A.2.1 Change Default Port and Default WebSpeed Broker/Server Name Values A–14
A.2.2 Minimize the PROPATH Entries A–15
A.2.3 Techniques To Minimize Unauthorized Access To WebSpeed Messengers A–15

A.3 Authenticating a Password Using SpeedScript A–16
A.3.1 Validating a Password Using SpeedScript A–16
A.4 Securing Data Transmissions Over the Internet A–18
A.4.1 Using Secure HTTP Protocol (HTTPS) A–18
A.4.2 Using Secure Sockets Layer (SSL) A–18
A.5 Maximizing WebSpeed Compatibility With Your Firewall A–19
A.5.1 WebSpeed TCP Port Details A–19
A.5.2 Important WebSpeed and Firewall Considerations A–20
A.5.3 Network Address Translation (NAT) and the WebSpeed Distributed Configuration A–21
A.5.4 Using the NameServer Client Port Range Value Settings With a Firewall A–25
A.6 Additional Security Information Available On the Internet A–26

B. State Management Using Connection Identifiers B–1
B.1 The Advantages Of Using SERVER-CONNECTION-ID B–2
B.2 Enabling Connection Identifiers B–3

C. Using Active Server Pages With WebSpeed C–1
C.1 Setting Up the WebSpeed ASP (WSASP) Messenger C–2
C.2 Accessing the WebSpeed ASP Web Page C–3
C.3 Enabling the WebSpeed ASP Example C–4

D. Using JavaScript With WebSpeed D–1
D.1 SpeedScript Versus JavaScript D–2
D.2 Using JavaScript Source Files D–3
D.3 Some JavaScript Examples D–4
D.3.1 A Combo Box With URL Buttons D–4
D.3.2 Browser Detection D–4
D.3.3 Printing From an HTML Page D–5

E. WebSpeed Dynamic Code-page Support E–1
E.1 Overview E–2
E.2 Determining the MIME Code-page Name E–3
E.3 Storing the MIME Code-page Name E–4
E.4 Accessing WEB-CONTEXT's HTML-CHARSET Attribute E–5
E.5 Converting Code-page Names Between MIME and Progress Formats . E–6
E.6 Additional Notes E–7

Figures

Figure A–1:	Agent Application Mode Field	A–4
Figure A–2:	Internal Administration Command - WSMAdmin Field	A–7
Figure A–3:	IP Address Field	A–9
Figure A–4:	List of IP Addresses	A–10
Figure A–5:	Debug Mode Field	A–12
Figure A–6:	Registration Mode Field	A–22
Figure B–1:	Using Progress Explorer To Change Messenger Properties	B–3
Figure B–2:	The WebSpeed Messenger Properties Dialog Box	B–4
Figure C–1:	The WebSpeed ASP Web Page	C–3
Figure C–2:	wsbroker1 Properties Dialog Box	C–4

Tables

Table 1–1:	WebSpeed Environment Variables	1–5
Table 1–2:	Collations Added To Version 3.1	1–28
Table 1–3:	Code-page Conversions Added To Version 3.1A	1–29
Table 2–1:	WebSpeed Version 3.1D Documentation Changes	2–3
Table A–1:	Agent Application Mode Descriptions	A–3
Table A–2:	Comparison Of the WebSpeed Messenger Administration Utility Default Settings	A–6
Table A–3:	How the Production Mode Effects the Debug Mode Values	A–13
Table A–4:	How the Development Mode Effects the Debug Mode Values	A–13
Table A–5:	Typical Default Port Numbers To Change	A–14
Table A–6:	Network Communications Requirements	A–19
Table A–7:	Registration Mode Connection - Progress Explorer	A–23
Table A–8:	RegistrationMode Connection - ubroker.properties File	A–24
Table A–9:	Supplemental Security-related Progress White Papers	A–26
Table E–1:	MIME and Progress Code-page Equivalents	E–3

Preface

Purpose

This bulletin describes the new and modified features of WebSpeed Versions 3.1A, 3.1B, 3.1C, and 3.1D. It also contains some supplemental documentation on features already described in the WebSpeed documentation set.

Audience

This bulletin is for WebSpeed application developers who are upgrading their licenses from WebSpeed Version 3.0B, 3.1A, 3.1B or 3.1C to WebSpeed Version 3.1D.

Organization Of This Manual

This bulletin is divided into the following chapters:

Chapter 1, “Overview Of New and Revised Features”

This chapter provides a summary of the product features that were either new or modified in WebSpeed Versions 3.1A, 3.1B, 3.1C and 3.1D. It provides references to the WebSpeed and Progress documentation sets where you can find more information on the product features.

Chapter 2, “Documentation”

This chapter describes the different formats in which WebSpeed documentation is available. It also provides an overview of documentation changes in WebSpeed Versions 3.1A, 3.1B, 3.1C and 3.1D.

Appendix A, “WebSpeed Security Issues”

This appendix highlights some of the basic security issues with regard to WebSpeed.

Appendix B, “State Management Using Connection Identifiers”

This appendix describes how you can use unique connection identifiers to maintain context between web requests.

Appendix C, “Using Active Server Pages With WebSpeed”

This appendix contains information for developers who want to use Microsoft’s Active Server Page (ASP) technology in conjunction with their WebSpeed applications.

Appendix D, “Using JavaScript With WebSpeed”

This appendix contains information for developers who want to use JavaScript in conjunction with their WebSpeed applications.

Appendix E, “WebSpeed Dynamic Code-page Support”

This appendix describes how to incorporate dynamic code-page support into WebSpeed applications.

How To Use This Manual

This bulletin is an overview of the feature and documentation changes in WebSpeed Versions 3.1A, 3.1B, 3.1C and 3.1D. It is also a guide that tells you where to find more information on new features.

The appendixes contain supplemental documentation on some topics that concern WebSpeed developers.

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 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

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 Installation and Configuration Guide

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

WebSpeed Developer's Guide

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

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

Overview Of New and Revised Features

This chapter lists the new and revised features implemented in WebSpeed Version 3.1D. In addition, it summarizes the changes that were made in WebSpeed Versions 3.1A, 3.1B, and 3.1C. Where applicable, it refers you to more extensive documentation on new and revised features.

New and revised features are grouped by release in the following sections:

- Changes In Version 3.1D
- Changes In Version 3.1C
- Changes In Version 3.1B
- Changes In Version 3.1A

Each section also describes where the feature is documented.

1.1 Changes In Version 3.1D

This section describes new and revised features in WebSpeed Version 3.1D in the following areas:

- Dynamic Code Page Support
- AdminServer Security Enhancement
- Optional NameServer Enhancement
- New Environment Variables For the WebSpeed Broker
- New WebTools
- Enhanced Diagnostic Logging For WebSpeed Messenger
- The ping.p, reset.p, and debug.p Utilities Moved

1.1.1 Dynamic Code Page Support

Progress Version 9.1D and WebSpeed Version 3.1D offer dynamic code-page support. This feature lets a Web browser running one code page and a WebSpeed Agent running a different, but compatible, code page exchange data without corrupting it.

With dynamic code-page support, a single Agent can handle requests from multiple browsers each running a different code page compatible with the Agent's code page. This can reduce the number of Agents required.

In fact, a single Agent running the UTF-8 encoding of Unicode (a standard whose goal is to map all characters of all the world's languages to a single code page) can handle Web requests from multiple browsers each running a code page compatible with Unicode. This can reduce the number of Agents required.

For more information on this feature, see Appendix E, "WebSpeed Dynamic Code-page Support."

Manuals:	*WebSpeed Installation and Configuration Guide*
	Progress Version 9 Product Update Bulletin

1.1.2 AdminServer Security Enhancement

Progress Version 9.1D and WebSpeed Version 3.1D have optional functionality that allows access to the AdminServer based on a user's membership in a group that has the appropriate privileges to perform AdminServer operations. This level of security is known as *authorization*.

Prior versions of Progress and WebSpeed supported *authentication*. Authentication allows validation of a user's identity by requiring a username and password. Authorization defines what users can do after they successfully log in. If authorization is enabled, users must belong to a group that has privileges to perform AdminServer operations. WebSpeed developers, for example, usually require access to the AdminServer for controlling and configuring databases, brokers, agents, NameServers, and messengers. Those developers would have to belong to a group that is defined to allow them to perform AdminServer tasks.

Authorization is optional. You choose to enable or disable authorization during installation. If you choose to disable authorization, the AdminServer functions as it did in prior versions of Progress and WebSpeed. There is no authorization (group checking) of users. If you choose to enable authorization, the installation prompts for a list of groups that have privileges to perform AdminServer tasks. Creating groups and determining group membership should be done prior to the installation.

Manuals:	*Progress Version 9 Product Update Bulletin*
	WebSpeed Installation and Configuration Guide
	Progress Installation and Configuration Guide Version 9 for UNIX
	Progress Installation and Configuration Guide Version 9 for Windows

1.1.3 Optional NameServer Enhancement

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.

Manual:	*Progress Version 9 Product Update Bulletin*

1.1.4 New Environment Variables For the WebSpeed Broker

In WebSpeed Version 3.1D, WebSpeed Brokers and the Agents under their control can be modified by a number of predefined variables. These predefined variables affect a variety of conditions, including the development environment, logging, and database connections.

The following is a list of the new environment variables with a brief description of their functionality. They are enabled from the Progress Explorer Environment Variables dialog box. For more information on using these variables, see the online help in Progress Explorer.

Table 1–1: WebSpeed Environment Variables

Variable Name	Function
BATCH_INTERVAL	Set the amount of time agents sit idle before running the run-batch-object custom procedure.
COMPILE_ON_FLY	Compile source code at run time.
COMPILE_XCODE	Compile encrypted code.
DATABASES	Specify which databases to associate with a WebSpeed Agent.
DB_FAILOVER	Specify backup databases for the primary databases defined in DATBASES.
DB_GROUP	Define associated databases as groups.
DB_OBJECT	Associate code objects with databases.
LOG_PATH	Specify where to store log files.
LOG_TYPES	Specify what type of information to store in log files.
MULTI_DEV_PROPATH	Define unique PROPATH settings for multiple developers.
SESSION_COOKIE	Define a cookie that tracks anonymous users.
SESSION_PATH	Specify where to store session information.
STATE_AWARE_ENABLED	Enable state aware mode.
SUPER_PROC	Specify the super procedures to initialize.
WEB_RUN_PATH	Specify which pathnames are accessible via a URL.

1.1.5 New WebTools

WebSpeed Version 3.1D has two new WebTools, the Application Manager and the Data Browser. Both tools have been added to the WebSpeed Webtools menu.

The Application Manager contains the following utilities:

- **Servers** — a utility for managing the WebSpeed Brokers, database servers and other servers.
- **CodePath** — a utility for defining which directories to include in your code search path. It also allows you to compile all or selected source code files.
- **R-Code Library** — a utility that allows you to package your r-code into procedure library (`.pl`) files.

The Data Browser allows you to work with the actual content of a database and includes the following utilities:

- **Dictionary Report** — a utility that lists fields and indexes, along with their types, formats, labels, and column labels.
- **Record Browse** — a utility that allows you to scroll through records in a table. You can also create, delete, and modify records.

The Data Browser can also show information about virtual system tables and hidden metaschema tables.

For more information on how to use the Application Manager and Data Browser see the online help, which is accessible from the Help button at the top of each tool.

1.1.6 Enhanced Diagnostic Logging For WebSpeed Messenger

With WebSpeed Version 3.1D, it is possible to enable enhanced diagnostic logging for the WebSpeed Messenger. If diagnostic logging is enabled, the WebSpeed Messenger enters an error message into a log file (`msgr.log`) when an error occurs. If there are no errors, the log file is not created.

To enable enhanced diagnostic logging, you must edit the `ubroker.properties` file. Go to the WebSpeed Messenger Properties section and find the properties related to logging for the type of Messenger that you are using (CGIIP, ISAPI, or NSAPI).

The `ubroker.properties` file is documented so that you can understand how to set logging levels and how to manage log files.

1.1.7 The ping.p, reset.p, and debug.p Utilities Moved

Because of security considerations, several utilities are now implemented as external files and are removed from `web/objects/web-disp.p`. The files `debug.p`, `ping.p` and `reset.p` are now in the `webutil` directory. You must now use the `webutils` directory in the URL query string that calls these utilities. For example, the URL to run ping would look similar to the following:

```
http://localhost/cgi-bin/cgiip.exe/WService=wsbroker1/webutil/ping.p
```

1.2 Changes In Version 3.1C

This section describes new and revised features in WebSpeed Version 3.1C in the following areas:

- Java Requirements
- Enhancements To the WebSpeed Error Customization Utility

1.2.1 Java Requirements

Java requirements changed. In general, beginning with Version 3.1C, WebSpeed requires Version 1.3.0 of JDK/JRE. Java requirements by platform are listed in the *WebSpeed Installation and Configuration Guide*

1.2.2 Enhancements To the WebSpeed Error Customization Utility

The WebSpeed Error Customization Utility has been enhanced so that you can customize all Progress error messages. The WebSpeed Error Customization Utility is now documented in the *WebSpeed Installation and Configuration Guide*.

1.3 Changes In Version 3.1B

This section describes the features that were new and revised for Version 3.1B in the following areas:

- Saving Changes To the ubroker.properties File
- Customizing Error Messages
- OS Locale Startup Parameter
- Development Tools
- SpeedScript

1.3.1 Saving Changes To the ubroker.properties File

In Version 3.1B, you can retain changes made to the previous version's `ubroker.properties` file. To do this, replace the default properties file in the installed properties directory with the old properties file. When you start the AdminServer for the first time, a properties file conversion utility automatically runs.

Manuals:	*Progress Installation and Configuration Guide Version 9 for Windows* *Progress Installation and Configuration Guide Version 9 for UNIX* *Progress on the Web*

1.3.2 Customizing Error Messages

In Version 3.1B, the WSMAdmin tool was extended to allow you to customize certain Progress error messages. However, in Version 3.1C the WSMAdmin tool was extended further to allow you to customize all Progress error messages. The WebSpeed Error Customization Utility is now documented in the *WebSpeed Installation and Configuration Guide*.

1.3.3 OS Locale Startup Parameter

Version 3.1B provided the Use OS Locale (–useOsLocale) startup parameter to set certain default localization options in WebSpeed Agents (Windows only) when you start them up. This parameter is now documented in *Progress Startup Command and Parameter Reference*.

1.3.4 Development Tools

Version 3.1B included changes to the following Progress development tools:

- ADM Tools
- AppBuilder
- Application Compiler
- PRO*Tools

ADM Tools

The following changes include new SmartObjects that support Internet connectivity:

- A Simple SmartContainer that has no run-time representation, useful when creating modules that require no user interface
- A new, dynamic SmartDataObject that reduces the client-side memory requirement
- A new SmartBusinessObject that integrates up to twenty SmartDataObjects, allowing comprehensive updates in a single server-side transaction
- A new SmartB2BObject that transforms 4GL data to and from XML for use in business-to-business messaging
- New SmartSender and SmartReceiver objects that you can extend to create and handle a variety of message types
- A new SmartRouter object that distributes incoming documents for processing

New SmartProducer and SmartConsumer objects that provide a messaging interface to the Progress SonicMQ system.

Manual:	*Progress on the Web*

The following changes include general enhancements to support the ADM 2:

- Changes in the behavior of three super-procedure routines to support the dynamic object
- New naming convention for SmartObject files
- New property ObjectName in all SmartObjects

Manual:	*Progress on the Web*
Online Help:	ADM Help

AppBuilder

The changes to AppBuilder in Version 3.1B were:

- Tool icons for the new ADM SmartObjects added to the Object Palette
- The ability to indirectly invoke the new XML Schema Mapping Tool by opening an XML schema or mapping file

Manual:	*Progress on the Web*

Application Compiler

In Version 3.1B, the Application Compiler supports a new option to compile r-code to the minimum size allowable for execution. This more compact code eliminates r-code segment information needed only by the Application Debugger and the Open Client Toolkit.

Manuals:	*Progress Application Development Environment — Getting Started*
	Progress on the Web
Online Help:	Application Compiler Help

PRO*Tools

For Version 3.1B, the XML Schema Mapping PRO*Tool was added to the PRO*Tools palette. The XML Schema Mapping PRO*Tool is useful for creating the mapping files required by the new SmartB2BObject.

Manual:	*Progress on the Web*
Online Help:	PRO*Tools Help

1.3.5 SpeedScript

Version 3.1B included the following changes to SpeedScript:

- Enhanced 4GL to support Progress on the Web
- Modified PROPATH to support URL pathnames for use by the SEARCH function
- Modified SEARCH function to support a URL pathname for the file argument
- New MIN–SIZE option on the COMPILE statement to reduce r-code size
- New CLIENT–TYPE attribute on the SESSION system handle
- New AVAILABLE attribute and DECIMALS attribute on the buffer-field object handle
- New HAS–RECORDS attribute and modified ADD–NEW–FIELD( ) method on the temp-table object handle

Manuals:	*Progress on the Web*
	Pocket WebSpeed

1.3.6 Virtual Directories

WebSpeed Version 3.1B included version-specific virtual directories for the WebSpeed WorkShop and Example files. This makes the coexistence of different versions of WebSpeed on your system simpler.

1.4 Changes In Version 3.1A

This section describes the features that were new and revised in WebSpeed Version 3.1A in the following areas:

- Installation
- Virtual Directories
- Administration Tools
- Development Tools
- SpeedScript
- Internationalization
- XML
- Sockets

1.4.1 Installation

Version 3.1A included the following changes that might affect how you install WebSpeed:

- The EDOC Installation utility now installs the DynaText 4.1A Viewer.
- A new feature allows users to add components or subcomponents to existing Progress Version 9.1A and WebSpeed 3.1A and later installations without having to enter any data other than the required components or subcomponents.
- On NT, the Installation utility now installs the Microsoft Management Console (MMC) if you are using one of the six language versions available with the Installation utility, and if you do not already have MMC 1.1 or later installed on your system. The Progress Explorer now operates as a snap-in component of the MMC. The Installation utility calls the Microsoft-supplied install program to install Microsoft Java and the MMC.

- The installation utility now provides a silent mode for installing on UNIX platforms (this feature was already available on Windows platforms). You can use the batch install feature to launch the installation from your own install utility, and WebSpeed will be installed silently during your install.

1.4.2 Administration Tools

WebSpeed Version 3.1A included the following changes:

- The WebSpeed Messenger Administration (WSMAdmin) tool has a new security feature. You can restrict the tool so that only users with certain IP addresses can access it. You can create a list of allowed IP addresses in the Progress Explorer or by manually editing the `ubroker.properties` file.

- The Progress Explorer tool, the graphical user interface that provides an easy way for you to manage Progress servers, is now a snap-in for the Microsoft Management Console (MMC). The Progress Explorer tool runs on the Windows 95, Windows 98, Windows 2000, and Windows NT platforms. This release provides online HTML help for the Progress Explorer.

- The following changes were made to the PROLIB utility to support memory-mapped procedure libraries:

  - A new -makeshared parameter lets you generate a memory-mapped procedure library from a standard procedure library.

  - In addition to existing library information, the -list parameter now displays the format of the library. The valid library formats are STANDARD and MAPPED.

Manuals:	*Progress Programming Handbook*
	Progress Client Deployment Guide
	WebSpeed Installation and Configuration Guide

- The following startup parameters were added:

New Parameter	Syntax	Purpose
Async Queue Size	-asyncqueuesize *n*	Specify the total number of bytes allowed for asynchronous request buffers allocated for a client.
Initial Value Segment No Convert	-isnoconv	Disables conversion of INIT and :U strings when they contain non-ASCII characters. Use if the conversion garbles preconverted international characters or causes the text segment to grow too large.
Fractional Separator	-numdec *numeric-value*	Changes the character that represents a number's decimal point in formatted text.
Thousands Separator	-numsep *numeric-value*	Changes the character that represents a number's thousands separator in formatted text.
Reread Nolock	-rereadnolock	Directs Progress to re-read the record from the database, even if the record is already in a buffer.
Four Digit Year Default	-yr4def	Outputs a four-digit year from EXPORT, MESSAGE and PUT UNFORMATTED statements that might use a two-digit year.

Manuals: *Progress Startup Command and Parameter Reference*

- The following startup parameters became obsolete:

Obsolete Parameter	Syntax	Purpose
Character Set	-charset	Identifies the code page that WebSpeed uses in memory. Replaced by -cpinternal.
Cyclic Redundancy	-crc	Specifies using Cyclic Redundancy Checking to ensure integrity between the database schema and compiled code.
Heap Size	-hs	Specifies the heap space on System V UNIX machines that use shared memory.
Network Version	-Nv	Specifies the version number of the connection protocol.
Page Writer Queue Delay	-pwqdelay	Determines how often the asynchronous page writer (APW) scans the page writer queue.
Page Writer Queue Minimum	-pwqmin	Determines the minimum number of buffers required on the page writer queue before the APW writes them to disk.
Page Writer Scan	-pwscan	Determines the number of buffers that an APW scans during a buffer scan cycle.
Page Writer Scan Delay	-pwsdelay	Determines the number of seconds delay between each APW scan of the buffers
Page Writer Maximum Buffers	-pwwmax	Determines the maximum number of modified buffers to write to disk during a buffer scan cycle
Stream	-stream	Specifies the code page for stream I/O. Replaced by -cpstream.
SYBASE	-SYBt	Sets the time of a SYBASE client timeout in seconds.

Obsolete Parameter	Syntax	Purpose
Time Stamp	-tstamp	Uses time stamp information rather than cyclic redundancy (CRC) to enforce consistency between r-code and a schema
Extended Alphabet Support	-xc	Specifies the collation table within the `convmap.cp` file.

1.4.3 Development Tools

Version 3.1A included the following changes that affect how you develop WebSpeed applications:

- The WebTools interface is extended to include:
  - A code editor
  - A tool for entering operating system commands
  - A display of the performance statistics that are contained in a database's Virtual System Tables
- Improvements to the WebSpeed Wizards include:
  - Linking and passing data to SmartDataObjects
  - Specifying instance properties for SmartDataObjects
  - Two new instance properties for SmartDataObjects

In addition, Version 3.1A included changes to the following tools that support WebSpeed development:

- ADM Tools
- AppBuilder
- Pro*Tools
- Windows Help

ADM Tools

The following Progress ADM Tools were new or revised in WebSpeed Version 3.1A:

- **SmartDataObject** — Three significant revisions have been made to SmartDataObjects:
  - You can define temp-tables for SmartDataObjects that contain subsets of the fields in the database tables that they will access.
  - They can run statelessly.
  - They now have four transaction-level validation entry points instead of only one.

Manuals:	*Progress ADM 2 Guide* *Progress ADM 2 Reference* *Progress AppBuilder Developer's Guide*

- **Extending the ADM** — Version 3.1A provided two new tools for developing extensions to the ADM; that is, for extending the ADM class hierarchy with new or customized classes. (In Version 3.0, you did this by copying and modifying existing class files.) The two new tools are:
  - Custom class files. The ADM provides a set of these files for each ADM class. You use them as the basis for a customized ADM class that is modeled on an existing class.
  - The ADM Class tool. This AppBuilder tool creates a set of ADM class files (standard and custom) according to your specifications. You use the standard class files that the tool creates as the basis for a new ADM class.

  While each of these new ADM features provides a set of class files for a customized or new class, these class files do not completely define the class and cannot be used as is. You must modify them with the application logic that drives the class: properties, super procedures, and so forth.

Manuals:	*Progress ADM 2 Guide* *Progress ADM 2 Reference*

AppBuilder

The AppBuilder had the following new or enhanced features added in Version 3.1A:

- AppBuilder reports on database connections when you choose Options→ Preferences→ WebSpeed→ Test.
- If your WebSpeed installation runs under a development license, AppBuilder editors include the following advanced editing features:
  - Color coding
  - Embedded color coding
  - Syntax expansion
  - Alias expansion
  - Syntax indenting
  - SmartPaste

NOTE: The advanced editing features are enabled by default. The advanced editing features do not support bi-directional typing on Hebrew and Arabic Windows operating systems. You can turn the advanced editing features on or off to meet your needs.

- You have the option to associate a separate Section Editor window with each open workspace. You can also continue to use a single window that shifts context according to which workspace has focus.
- The Section Editor allows you to leave open the windows from which you paste certain tokens into your source code.

- The File menu includes the following:
  - A Save All option that you use to save structured procedures. Structured procedures are those procedures being maintained by AppBuilder.
  - A Most Recently Used file list option. You can enable or disable this option. If you enable this option, you can use the File menu in the AppBuilder to select and open a file from a list of most recently used files. You use the Options→ Preferences option on the AppBuilder menu to specify the number of files you want to appear in this list.
  - A Print option that allows you to print either the section that is current in the Section Editor, or the entire current procedure. Qualities such as page length and immediate versus prompted printing are Preference items.
- The Tools menu includes an item (ADM) that, when selected, opens a dialog box to begin the process of creating a new ADM class.
- The SmartDataObject instance properties include a checkbox for forcing stateful (persistent-connection) operation. The default mode is stateless (connectionless) unless the AppServer is configured to run with bound connections.

Manuals:	*Progress AppBuilder Developer's Guide* *Progress Application Development Environment — Getting Started*

PRO*Tools

The PRO*Tools palette includes the following new or enhanced applets:

- PRO*Spy for Version 9 is a new applet for debugging applications containing ADM2 SmartObjects.

Online Help:	PRO*Tools Help

- The AppServer Session Information PRO*Tool is a new applet that provides information about a running AppServer session defined with the AppServer Partition Deployment PRO*Tool.

Manual:	*Building Distributed Applications Using the Progress AppServer*
Online Help:	PRO*Tools Help

- The Procedure Object Viewer lists any persistent super procedures associated with a selected persistent procedure.

Online Help: PRO*Tools Help

- The Database Connections PRO*Tool is now modal and displays enhanced schema information, including table, field, and index details.

Online Help: PRO*Tools Help

Windows Help

- Progress Windows help is now modular. All installed Windows help systems appear in a combined table of contents, index, and full text search. Windows help systems that are not installed on a user's computer do not appear in the combined table of contents, index, or full text search.

Online Help: Progress Help

1.4.4 SpeedScript

WebSpeed Version 3.1A included the following SpeedScript enhancements:

- New asynchronous request object and the following new and modified 4GL elements that support it:
  - New ASYNC-REQUEST-COUNT attribute
  - New CANCELLED attribute
  - New CANCEL-REQUESTS( ) method
  - New COMPLETE attribute
  - Modified DISCONNECT( ) method
  - New ERROR attribute
  - New EVENT-PROCEDURE attribute
  - New EVENT-PROCEDURE-CONTEXT attribute
  - New FIRST-ASYNC-REQUEST( ) method
  - New LAST-ASYNC-REQUEST( ) method
  - Modified NAME attribute
  - Modified NEXT-SIBLING attribute
  - New PERSISTENT-PROCEDURE attribute
  - Modified PREV-SIBLING attribute
  - Modified PRIVATE-DATA attribute
  - New PROCEDURE-COMPLETE event
  - New PROCEDURE-NAME attribute
  - New QUIT attribute
  - Modified RUN statement
  - Modified SELF system handle

  - Modified SERVER attribute
  - Modified server object handle
  - New STOP attribute
  - Modified TYPE attribute
  - Modified WAIT-FOR statement

- New temp-table object and the following new and modified SpeedScript elements that support it:
  - New CREATE TEMP-TABLE statement
  - New CLEAR( ) method
  - New TEMP-TABLE-PREPARE( ) method
  - New CREATE-LIKE( ) method
  - New ADD-FIELDS-FROM( ) method
  - New ADD-NEW-FIELD( ) method
  - New ADD-LIKE-FIELD( ) method
  - New ADD-NEW-INDEX( ) method
  - New ADD-INDEX-FIELD( ) method
  - New ADD-LIKE-INDEX( ) method
  - Modified NAME attribute
  - New DEFAULT-BUFFER-HANDLE attribute
  - New UNDO attribute
  - New PREPARED attribute
  - New PRIMARY attribute
  - Modified DEFINE PARAMETER statement

- Buffer object and Buffer-field object enhancements to support the temp-table object:
  - Modification to CREATE BUFFER statement to support using a temp-table as a table-name
  - New TABLE-HANDLE attribute
  - Modified NEXT-SIBLING attribute
  - New BUFFER-COPY( ) method
  - New BUFFER-COMPARE( ) method
  - New INDEX-INFORMATION( ) method
  - New RAW-TRANSFER( ) method
  - New PRIMARY attribute
  - Modified VALIDATE-EXPRESSION attribute
  - New FIRST-BUFFER attribute
- Super procedure enhancements to support modification of the super procedure search order:
  - Modified ADD-SUPER-PROCEDURE( ) method
- Enhanced print capabilities:
  - Modified PRINTER-NAME attribute for the SESSION system handle (Windows only)
  - New NUM-COPIES option for the OUTPUT TO PRINTER statement (Windows only)
  - New LANDSCAPE and PORTRAIT options for the OUTPUT TO PRINTER statement (Windows only)
  - New COLLATE option for the OUTPUT TO PRINTER statement (Windows only)
  - New LANDSCAPE and PORTRAIT options for the SYSTEM-DIALOG PRINTER-SETUP statement (Windows only)

- Miscellaneous SpeedScript enhancements:
  - PROCESS EVENTS statement now works for WebSpeed
  - Modified DATE function to support XML and socket input
  - Modified EXPORT statement to allow MEMPTR data type to be exported to a 4GL stream
  - Modified IMPORT statement to allow MEMPTR data type to be imported from a 4GL stream
- Support for UNIX shared libraries:
  - Modified PROCEDURE statement to support declaring a routine in a UNIX shared library
  - Modified RUN statement to support executing a routine in a UNIX shared library
- Enhancements to support Internationalization:
  - New COMPARE function
- New COLLATE option for the FOR statement, the OPEN QUERY statement, and the PRESELECT phrase.

Manuals:	*Progress Language Reference*
	Pocket WebSpeed

1.4.5 Internationalization

WebSpeed Version 3.1A provided several new or revised features related to internationalization:

- Support for international numeric formats
- Additions to SpeedScript
- Code pages and XML
- Added collations and code-page conversions

Additional Support for International Numeric Formats

Version 3.1A provided additional support for international numeric formats:

- The Fractional Separator (-numdec) startup parameter sets the character that separates the integer portion and the fractional portion of a decimal value.
- The Thousands Separator (-numsep) startup parameter sets the character that separates each group of three digits in the integer portion of a number.

Manuals:	*Progress Startup Command and Parameter Reference*
	Pocket WebSpeed

- The value of NUMERIC-FORMAT can now be a character string consisting of the thousands separator followed by the fractional separator.
- The NUMERIC-SEPARATOR attribute sets the character that separates each group of three digits in the integer portion of a number.
- The NUMERIC-DECIMAL-POINT attribute sets the character that separates the integer portion and the fractional portion of a decimal value.
- The SET-NUMERIC-FORMAT() method sets the NUMERIC-SEPARATOR and NUMERIC-DECIMAL attributes to arbitrary values simultaneously.

Manuals:	*Progress Language Reference*
	Pocket WebSpeed

Additions to SpeedScript

WebSpeed 3.1A included additions to SpeedScript that apply to internationalization:

- The COMPARE function compares two strings, using one of several methods (“RAW,” “CASE-SENSITIVE,” “CASE-INSENSITIVE,” or “CAPS”) and the collation table you specify.
- The COLLATE option of the FOR statement, the OPEN QUERY statement, and the PRESELECT phrase computes the collation value of a string using one of several methods (“RAW,” “CASE-SENSITIVE,” “CASE-INSENSITIVE,” or “CAPS”) and the collation table you specify.

Manuals:	*Progress Language Reference* *Pocket WebSpeed*

XML Code Page Support

XML documents can be encoded using any of a wide a variety of character encodings.

Manual:	*Progress External Program Interfaces*

Collations and Code-page Conversions

Version 3.1A made a number of changes to collations and code-page conversions described in the following sections.

Table 1–2 lists the collations added to Version 3.1.

Table 1–2: Collations Added To Version 3.1

Code Page Name	Collation Name
1250	CROATIAN
1254	TURKISH-COMP
1256	ARABIC9
1257	LITHUANIAN
620-2533	BASIC

NOTE: WebSpeed Version 3.1A provided a new Thai collation, BASIC. This collation, available for the 620-2533 code page, follows the rules of Thai sorting more closely. The empty database in the `DLC/prolang/tai` directory uses this collation. Progress Software Corporation recommends that you use this new collation. See Release Note 091A-00550 for more information.

For more information on converting databases from an older collation to a newer collation, see the chapter on character processing tables in the *Progress Internationalization Guide*.

Renamed Collation

Version 3.1A renamed the THAI collation of code page 620-2533 to THAI-OLD. It is still available through the `thai-old.df` file in the `DLC/prolang/tai` directory.

New Code-page Conversions

Table 1–3 lists the code-page conversions added to Version 3.1A.

Table 1–3: Code-page Conversions Added To Version 3.1A

Source Code Page	Target Code Page
1257	UTF-8
UTF-8	1257

Manual:	*Progress Internationalization Guide*

New Initial Value Segment No Convert (-isnoconv) Startup Parameter

The -isnoconv option disables conversion of INIT and :U strings that contain non-ASCII characters.

By default, INIT and :U strings are codepage converted if they contain non-ASCII characters. It is possible that compiling existing code may result in larger r-code segments and compilation may fail due to exceeding segment limits.

Manual:	*Progress Startup Command and Parameter Reference*

Memory-mapped R-code and Automatic Code-page Conversion

WebSpeed Version 3.1A offers memory-mapped r-code, which reduces the amount of memory required to run multiple WebSpeed and Progress processes. Because memory-mapped files are shared, they are read-only, which means that WebSpeed cannot convert the code page as the file is loaded. So if the r-code code page (-rcodein) and the internal memory code page (-cpinternal) are different, WebSpeed reports an error and stops.

Manual:	*Progress Programming Handbook*

1.4.6 XML

This section specifies language extensions made to SpeedScript that enable WebSpeed applications to use XML through the integration of a third-party DOM parser. These extensions are:

- The new X-document object and the following new and modified SpeedScript elements that support it:
  - New CREATE X-DOCUMENT statement
  - New ENCODING attribute
  - Modified NAME attribute
  - New NUM-CHILDREN attribute
  - New SUBTYPE attribute
  - Modified TYPE attribute
  - Modified UNIQUE-ID attribute
  - New APPEND-CHILD( ) method
  - New CREATE-NODE( ) method
  - New GET-CHILD( ) method
  - New GET-DOCUMENT-ELEMENT( ) method
  - New IMPORT-NODE( ) method
  - New INSERT-BEFORE( ) method
  - New LOAD( ) method
  - New REMOVE-CHILD( ) method
  - New REPLACE-CHILD( ) method
  - New SAVE( ) method

- The new X-noderef object and the following SpeedScript elements that support it:
  - New CREATE X-NODEREF statement
  - New ATTRIBUTE-NAMES attribute
  - New CHILD-NUM attribute
  - Modified NAME attribute
  - New NODE-VALUE attribute
  - New NUM-CHILDREN attribute
  - New OWNER-DOCUMENT attribute
  - New SUBTYPE attribute
  - Modified TYPE attribute
  - Modified UNIQUE-ID attribute
  - New APPEND-CHILD( ) method
  - New CLONE-NODE( ) method
  - New DELETE-NODE( ) method
  - New GET-ATTRIBUTE( ) method
  - New GET-CHILD( ) method
  - New GET-PARENT( ) method
  - New INSERT-BEFORE( ) method
  - New REMOVE-ATTRIBUTE( ) method

- New REMOVE-CHILD( ) method
- New REPLACE-CHILD( ) method
- New SET-ATTRIBUTE( ) method

Manuals:	*Progress Language Reference* *Progress External Program Interfaces* *Pocket WebSpeed*

1.4.7 Sockets

WebSpeed Version 3.1A provided SpeedScript support for accessing TCP/IP sockets directly rather than having to use the HLC Interface. SpeedScript supports both client side and connection-oriented server side implementations. This section describes language extensions to SpeedScript that enable WebSpeed applications to use TCP/IP sockets:

- New socket object and the following new and modified SpeedScript elements that support it:
  - New CREATE SOCKET statement
  - Modified DELETE OBJECT statement
  - New BYTES-READ attribute
  - New BYTES-WRITTEN attribute
  - New REMOTE-HOST attribute
  - New LOCAL-HOST attribute
  - Modified PRIVATE-DATA attribute
  - New REMOTE-PORT attribute
  - New LOCAL-PORT attribute
  - Modified TYPE attribute
  - Modified NEXT-SIBLING attribute
  - Modified PREV-SIBLING attribute

  - Modified HANDLE attribute
  - Modified NAME attribute
  - Modified SENSITIVE attribute
  - Modified CONNECT( ) method
  - New SET-READ-RESPONSE-PROCEDURE( ) method
  - New READ( ) method
  - New WRITE( ) method
  - Modified DISCONNECT( ) method
  - Modified CONNECTED( ) method
  - New GET-BYTES-AVAILABLE( ) method
  - New SET-SOCKET-OPTION( ) method
  - New GET-SOCKET-OPTION( ) method

- New server socket object and the following new and modified SpeedScript elements that support it:
  - New CREATE SERVER-SOCKET statement
  - Modified DELETE OBJECT statement
  - Modified TYPE attribute
  - Modified PRIVATE-DATA attribute
  - Modified HANDLE attribute
  - Modified NAME attribute
  - Modified SENSITIVE attribute
  - New ENABLE-CONNECTIONS( ) method
  - New SET-CONNECT-PROCEDURE( ) method
  - New DISABLE-CONNECTIONS( ) method

- New socket events:
  - New READ-RESPONSE event
  - New CONNECT event
- Modified handles for socket support:
  - Modified SELF Handle
  - Modified SESSION System Handle
- Miscellaneous new and modified statements and functions for marshalling and unmarshalling data for socket support:
  - Modified LIST-EVENTS function
  - Modified LIST-QUERY-ATTRS function
  - Modified LIST-SET-ATTRS function
  - Modified LIST-WIDGETS function
  - Modified GET-DOUBLE function
  - Modified GET-FLOAT function
  - Modified GET-LONG function
  - Modified GET-SHORT function
  - Modified GET-UNSIGNED-SHORT function
  - Modified PUT-DOUBLE statement
  - Modified PUT-FLOAT statement
  - Modified PUT-LONG statement
  - Modified PUT-SHORT statement
  - Modified PUT-UNSIGNED-SHORT statement
  - Modified GET-STRING function
  - Modified PUT-STRING statement

  - New GET-BYTES function
  - New PUT-BYTES statement

- Miscellaneous new statements and functions for byte ordering and bit manipulation for socket support:
  - New SET-BYTE-ORDER statement
  - New GET-BYTE-ORDER function
  - New GET-BITS function
  - New PUT-BITS statement

Manuals:	*Progress Language Reference*
	Progress External Program Interfaces
	Pocket WebSpeed

2

Documentation

This chapter provides an overview of the changes in the WebSpeed documentation set, starting with Version 3.1D and going back through Version 3.1A. It also provides an overview of the available options for electronic documentation.

For information on other manuals in the Progress documentation set, see "Other Useful Documentation" in the "Preface."

The topics covered in this chapter include:

- Electronic Documentation (EDOC)
- Summary Of Version 3.1D Documentation Changes
- Summary Of Version 3.1C Documentation Changes
- Summary Of Version 3.1B Documentation Changes
- Summary Of Version 3.1A Documentation Changes

2.1 Electronic Documentation (EDOC)

This section provides a brief description of the electronic formats in which WebSpeed and other Progress documentation is available. For more information, see the *Electronic Documentation Configuration and Installation Guide*.

2.1.1 DynaText

Documentation in the SGML format is viewable through the DynaText browser. Both the browser and the WebSpeed manuals are provided on a separate CD that is shipped with Progress products.

The DynaText browser features powerful search capabilities, which enables searching across multiple books and multiple collections. only baloney

2.1.2 PDF

Documentation in the PDF format is provided on a separate CD that is shipped with Progress products. In addition, documentation in the PDF format is available on the following Progress Software Corporation web site:

```
http://www.progress.com/v9/documentation
```

You can view and print these PDF files using the Acrobat® Reader™, which you can download for free from Adobe® Software's Web site:

```
http://www.adobe.com/products/acrobat/readstep.html
```

Beginning with Version 3.1C, you can search across multiple PDF files.

2.1.3 Dynamic HTML

Documentation in the Dynamic HTML format is available on the Progress Software Corporation web site, (`http://www.progress.com/v9/documentation`).

You can use any HTML browser to access the entire documentation set from any computer that is connected to the internet. Dynamic HTML is provided through a server-side plug-in called DynaWeb. DynaWeb translates our SGML source books into HTML on the fly. The DynaWeb user interface and powerful search engine are similar to that of the DynaText SGML Viewer (provided with all Progress and WebSpeed products).

2.2 Summary Of Version 3.1D Documentation Changes

Table 2–1 lists the WebSpeed documentation components and how each has changed in Version 3.1D.

Table 2–1: WebSpeed Version 3.1D Documentation Changes *(1 of 2)*

Documentation Component	Summary of Changes
Getting Started with WebSpeed	• Updated the screen capture of the WebTools main page and also updated descriptions of the WebTools. • Updated several screen captures in the tutorial in Chapter 5. • Corrected syntax errors in HTML examples in Chapter 5. • Corrected URL error for corplogo.gif in the tutorial Chapter 5. • Corrected miscellaneous typos.
WebSpeed Installation and Configuration Guide	• Added section on dynamic code-page support. • Updated "Database Connection Limits" table. • Added information on shared installations on Windows networks. • Added information on the Product Update utility, the SHOWCFG utility and batch-mode installations. • Added information on enhanced AdminServer security. • Revised information in "Connecting to Mult-national Databases" section. • Updated "Java Requirements" section. • Added a chapter on configuration utilities, which includes documentation for the WebSpeed Error Customization Utility.
AppBuilder WinHelp	Created a new section for WebSpeed API functions and procedures.
Progress Explorer WinHelp	Added help for the new environment variables.

Table 2–1: WebSpeed Version 3.1D Documentation Changes *(2 of 2)*

Documentation Component	Summary of Changes
WebTools HTML Help	Added help for new Webtools (Application Manager and Data Browser.)
Pocket WebSpeed	Added HTML-CHARSET attribute to the list of WEB-CONTEXT handle attributes.
WebSpeed Developer's Guide	No changes.
WebSpeed Product Update Bulletin	• Added section on Version 3.1D changes. • Condensed "Feature Comparison" information for prior releases and moved to Chapter 1. • Updated documentation references in prior releases. • Added appendix on dynamic code-page support. • Moved documentation changes to Chapter 2. • Moved information from "Java Requirements" and "Unicode Databases" sections to *WebSpeed Installation and Configuration Guide*.

2.3 Summary Of Version 3.1C Documentation Changes

Pocket WebSpeed was updated to include new language elements and startup parameters.

The other books in the WebSpeed documentation set (*Getting Started with WebSpeed*, *WebSpeed Installation and Configuration Guide*, and *WebSpeed Developer's Guide*) were not revised for Version 3.1C.

However, the *WebSpeed Product Update Bulletin* (this manual) was revised to include more information on topics such as security, state management, Active Server Pages, and JavaScript. This information is contained in the appendixes to this manual.

2.4 Summary Of Version 3.1B Documentation Changes

Version 3.1B included bug fixes and some minor changes to both the online and hard-copy documentation since Version 3.1A.

2.5 Summary Of Version 3.1A Documentation Changes

This section describes the documents in the Version 3.0 documentation set that have been updated for Version 3.1A:

- *Getting Started with WebSpeed*

  The following changes were made to this book:

  – Chapter 2, "Introducing the AppBuilder and WebTools," has been revised to include information about the new Editor, OS Command, and Virtual System Tables WebTools. It has also been reorganized to focus solely on the AppBuilder and WebTools.

  – Chapter 3, "Setting Up the WebSpeed Development Environment," is new. It describes how to make a working copy of the Sports2000 database, how to start a server for the database using Progress Explorer, how to connect the AppBuilder to the database, and how to connect the WebSpeed Development Server to the database using Progress Explorer.

  – Chapter 4, "Sample WebSpeed Applications," is new. It contains an overview of the sample WebSpeed applications and instructions on how to run them.

  – Chapter 5, "Creating a WebSpeed Application," is reorganized. It focuses on creating the SportsPro Sales Advisor application.

- *WebSpeed Developer's Guide*

  The primary change to this book was a reorganization of its chapters. The following summarizes the reorganization and updates made to this book:

  - Chapter 1, "Introducing WebSpeed," includes only introductory information.
  - Chapter 2, "Overview of WebSpeed Tools," includes only information about the tools and utilities used in WebSpeed application development. It was revised to include information about the new Editor, OS Command, and Virtual System Tables WebTools.
  - Chapter 4, "Sample Web Objects," focuses on describing the sample Web object code that is included with WebSpeed. It was revised to include a description of a new example, `w-custdir.w`, which shows how to use HTML Mapping with a SmartDataObject.
  - Chapter 5, "SpeedScript," focuses on information about SpeedScript and Embedded SpeedScript.
  - Chapter 8, "Controlling WebSpeed Transactions," now contains all the information about Web object state.
  - The "Building Applications" chapter was eliminated, but the information that it contained was moved to other chapters.

- *WebSpeed Installation and Configuration Guide*

  - User feedback indicated that many users were using WebSpeed on networks that had both Windows and UNIX systems. For convenience, the Windows and UNIX versions of this document have been combined into a single document.
  - Material throughout the document has been rearranged to provide a clearer separation of the tasks involved in planning your installation, installing WebSpeed, and configuring WebSpeed.
  - Chapters 1 through 4 give an overview of the WebSpeed architecture and discuss planning your WebSpeed installation. Chapter 3 describes several sample WebSpeed configurations.
  - Chapters 5 and 6 discuss installing WebSpeed on Windows and UNIX systems.

- Chapters 7 through 10 discuss configuring and using your WebSpeed installation.
- Chapters 11 and 12 discuss issues involved in upgrading earlier versions of WebSpeed and whether coexistence is possible.

- *Pocket WebSpeed*
  - This book has been updated to include new language elements and startup parameters.
- *Welcome to WebSpeed*
  - Describes the books available on the Electronic Documentation CD-ROM and provides new information about Progress Software Corporation's licensing agreement, as well as descriptions of the product services Progress offers to support WebSpeed customers.

A

WebSpeed Security Issues

This chapter highlights some security issues related to WebSpeed. Contents include:

- Changing WebSpeed Applications From Development Mode To Production Mode
- Changing Additional Settings To Minimize Security Risks
- Authenticating a Password Using SpeedScript
- Securing Data Transmissions Over the Internet
- Maximizing WebSpeed Compatibility With Your Firewall
- Additional Security Information Available On the Internet

A.1 Changing WebSpeed Applications From Development Mode To Production Mode

Changing your WebSpeed applications from Development mode to Production mode allows you to minimize some basic system vulnerabilities.

To transition from Development mode to Production mode, you need to

- Set the Agent Application Mode To Production
- Disable the WebSpeed Messenger Administration Utility
- Establish the WebSpeed Messenger Administration Internet Protocol List (Optional)
- Check the Status Of the Debug Mode

You can access these properties from either the Progress Explorer, or the `ubroker.properties` file that resides in the `install-path\properties` directory.

A.1.1 Set the Agent Application Mode To Production

You must set the value of your Agent application mode to Production to prevent unwanted exposure of several WebSpeed components. The default value for this setting is Development. Therefore, you must take explicit action to set this value to Production before you deploy and run your WebSpeed applications.

Table A–1 details the functionality differences between the Production and Development modes.

Table A–1: Agent Application Mode Descriptions

When the Agent Application Mode is set to...	Then this property...
Production	• Prohibits Web objects specified in the URL with the path `install-path\src\web\` from being run. • Disables these utility commands: debug, reset, ping, and WebTools. • Restricts the utility command ping so that it only indicates that the WebSpeed Agent is available; it does not display any other information about the Agent's environment. • Ignores debugging via the URL (?debug=xxx). However, if the Debug mode is set to enable, then debugging using the URL is not ignored.
Development (default)	• Allows unlimited access to run Web objects specified in the URL with the path `install-path\src\web\`. • Enables these utility commands: ping, debug, reset, web-utility, and workshop. • Allows users to access specific development files, including the WebSpeed Workshop. • Enables debugging via the URL (?debug=xxx). However, if the Debug mode is set to disable, then debugging using the URL is not ignored.

Procedure Using the Progress Explorer Tool

To change the value of the Agent application mode from Development to Production using the Progress Explorer Tool, perform these steps:

1 ♦ Double click the WebSpeed folder on the left-hand side of the main Progress Explorer window.

2 ♦ Right click on the broker object.

3 ♦ Choose Properties on the shortcut menu. The broker Properties window displays.

4 ♦ Click on Agent→ General to display properties associated with the WebSpeed Agent. Note that the Agent application mode field displays two options: Development and Production.

5 ♦ Set the Agent application mode to Production as shown in Figure A–1.

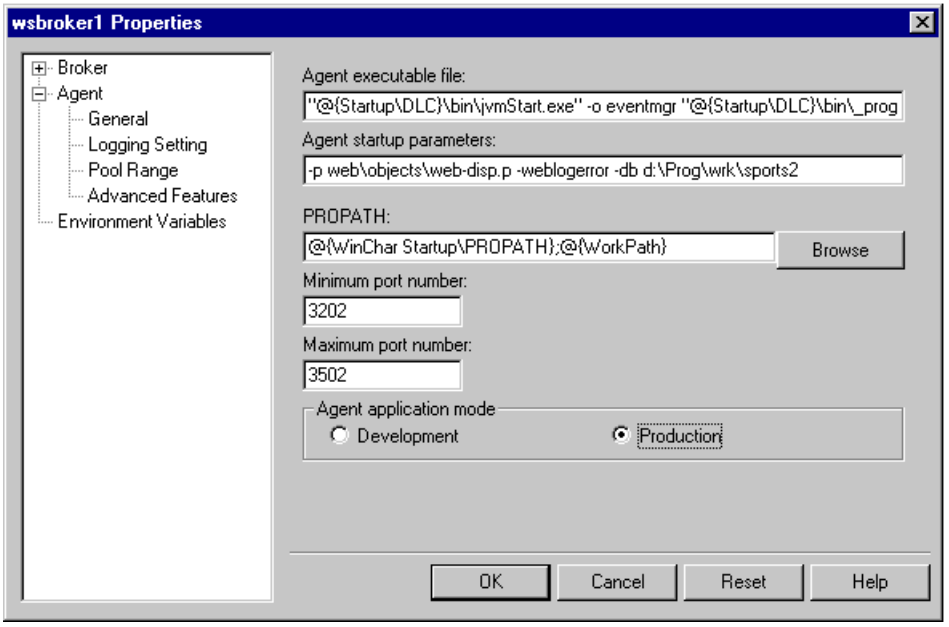

Figure A–1: Agent Application Mode Field

Procedure Using the ubroker.properties File

To change the value of the srvrAppMode property from Development to Production in the `ubroker.properties` file, perform these steps:

1 ♦ Determine if you need to set all brokers or just individual brokers to a Production mode.

The property values established in the various sections of the `ubroker.properties` file are governed by a parent-to-child inheritance relationship. For example, values set in the parent, the [UBroker] section, are inherited at the [UBroker.WS] level. Similarly, values defined at the [UBroker.WS] level are inherited as default values by each individual broker instance, established in unique [UBroker.WS.brokername] sections.

Alternatively, you can retain the property values established at the parent level and override only those individual values at the lower, child level that you need to change.

2 ♦ Search the `ubroker.properties` file for the srvrAppMode property in the section, or sections, that you need to change.

– To affect all brokers, search for srvrAppMode property in the [UBroker.WS] section.

– To affect one or more individual brokers, search for or add the srvrAppMode property in the individual [UBroker.WS.brokername] section.

3 ♦ Change the value of the srvrAppMode property to Production.

A.1.2 Disable the WebSpeed Messenger Administration Utility

You should prevent access to the WebSpeed Messenger Administration utility when you deploy your WebSpeed applications. It provides information about the NameServer, WebSpeed Broker, WebSpeed Agent, and other components of your system. It can also allow users to access static HTML files.

When enabled in the Development mode, this utility can help you debug problems in a WebSpeed configuration. However, if you leave this utility enabled when you move your system from a Development mode to a Production mode, you expose your system information to unauthorized access.

Ensure that you disable the WebSpeed Messenger Administration utility before you run your WebSpeed applications in Production mode.

NOTE: There is one exception to this guideline. If you are using this utility with the WebSpeed Messenger Administration Internet Protocol List option available in WebSpeed Version 3.1, you can leave the WebSpeed Messenger Administration utility enabled in Production mode.

See the following section, "Establish the WebSpeed Messenger Administration Internet Protocol List (Optional)", for details about using the WebSpeed Messenger Administration utility with the WebSpeed Messenger Administration Internet Protocol List option.

The default setting value for the WebSpeed Messenger Administration utility changed between WebSpeed Version 3.0 and WebSpeed Version 3.1A. Table A–2 compares the default value differences between these two product versions.

Table A–2: Comparison Of the WebSpeed Messenger Administration Utility Default Settings

In WebSpeed Version...	The default setting for WSMAdmin is...
3.0	Enabled; you can use the WebSpeed Messenger Administration utility as is to help debug problems in Development mode. When the application development is completed, you must disable this setting prior to changing to Production mode.
3.1A	Disabled; you must enable this property to use the WebSpeed Messenger Administration utility in Development mode. When the application development is completed, you must then disable this setting prior to changing to Production mode.

Procedure Using the Progress Explorer Tool

To disable the WSMAdmin utility for a WebSpeed Messenger from the Progress Explorer Tool, perform these steps:

1 ♦ Double click the Messengers folder on the left-hand side of the main Progress Explorer window.

2 ♦ Right click the specific messenger whose internal commands you want to disable.

3 ♦ Choose Properties from the shortcut menu. The properties window for the messenger you selected displays.

Figure A–2 shows the location of this field with the appropriate box checked on the CGIIP Properties window.

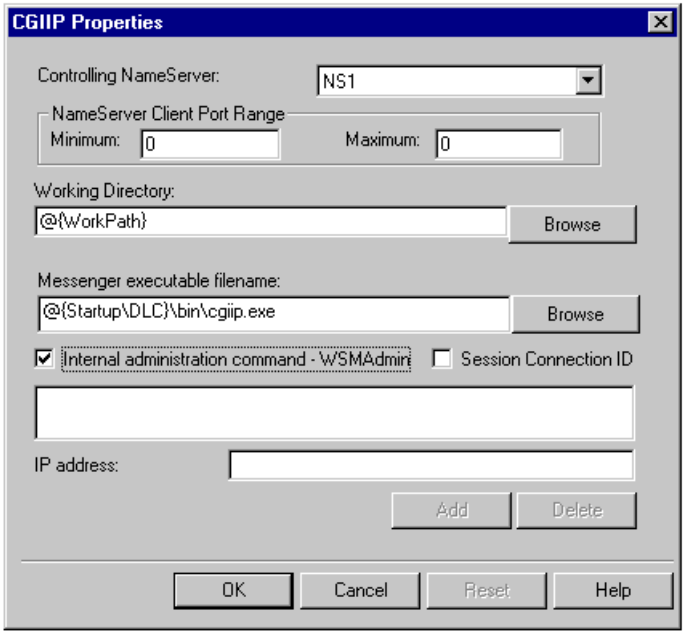

Figure A–2: Internal Administration Command - WSMAdmin Field

4 ♦ Deselect the check box next to the Internal Administration Command - WSMAdmin field.

5 ♦ If you are changing this value for either a WSISA or WSNSA Messenger, re-start your web server.

Verify that the Messenger Internal Commands are disabled. You should not be able to access the WSMAdmin utility and its features from a Web browser.

Procedure Using the ubroker.properties File

To disable the internal commands for a WebSpeed messenger from the `ubroker.properties` file, perform these steps:

1 ♦ Search the file for the AllowsMsngrCmds property. It is located in the properties section that is associated with the specific messenger type that your configuration is using.

For example, if you are using a CGIIP Messenger, the AllowsMsngrCmds property you must set is in the [WebSpeed.Messengers.CGIIP] section.

NOTE: Do not make changes in the [WebSpeed.Messengers] section because they will not affect this property.

2 ♦ Set the property value to 0 to disable the WebSpeed Messenger Administration utility commands.

Verify that the Messenger Internal Commands are disabled. You should not be able to access the WSMAdmin utility and its features from a web browser.

A.1.3 Establish the WebSpeed Messenger Administration Internet Protocol List (Optional)

The WebSpeed Messenger Administration Internet Protocol List option allows you to define a list of IP addresses that you can enable to access the WebSpeed Messenger's internal administrative commands. Any IP addresses not specified on this list are denied access to this utility.

You can use the Internet Protocol List option only if you enable the WebSpeed Messenger Administration utility. This utility maintains control over your list of IP addresses. Using the Internet Protocol List allows you the opportunity to debug specific applications remotely without exposing Web applications to unwanted access.

Procedure Using the Progress Explorer Tool

To add IP addresses using the Progress Explorer Tool, perform these steps:

1 ♦ Double click the Messengers folder on the left-hand side of the main Progress Explorer window.

2 ♦ Right click the specific messenger whose internal commands you want to enable.

3 ♦ Choose Properties from the shortcut menu. The properties window for the messenger you selected displays.

4 ♦ If the Internal administration command - WSMAdmin utility field is disabled, select the check box associated with this utility to enable it.

5 ♦ In the IP address field, type a unique IP address. Figure A–3 shows the location of this field with a sample value entered.

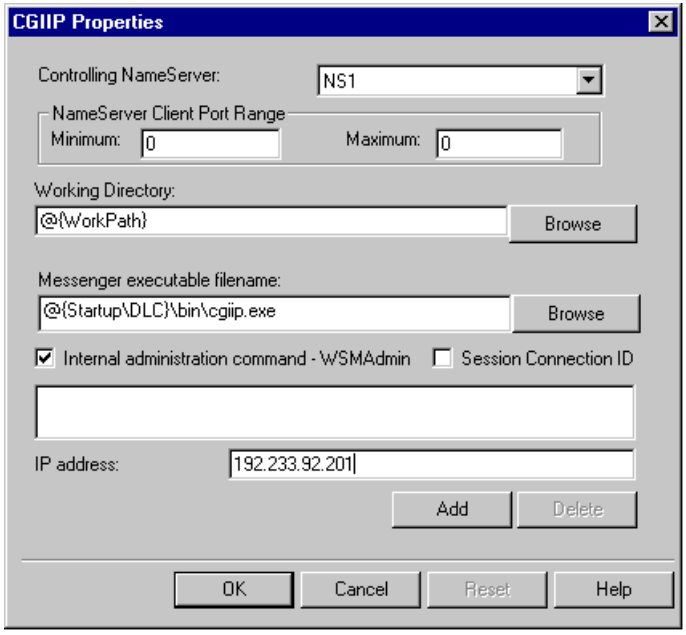

Figure A–3: IP Address Field

6 ♦ Press the Add button. The IP address you added now displays in the large text box above the IP address field.

7 ♦ For each additional IP address that you add, repeat steps 5 and 6.

Figure A–4 shows additional addresses added to the list.

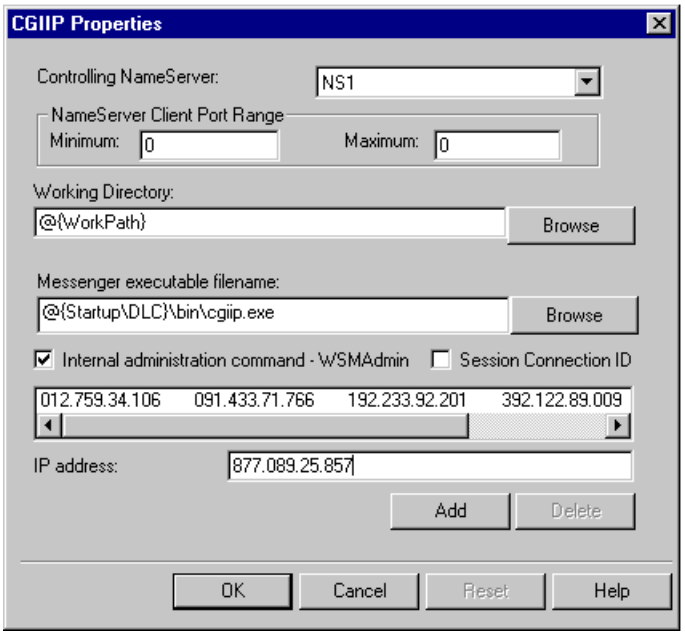

Figure A–4: List of IP Addresses

8 ♦ Click the OK button when you have added all address entries.

Procedure Using the ubroker.properties File

To define specific IP addresses using the `ubroker.properties` file, perform these steps:

1 ♦ Search the file for the AllowsMsngrCmds property. It is located in the messenger properties section that is associated with the specific messenger type that your configuration is using.

For example, if you are using a CGIIP Messenger, you must set or add the AllowsMsngrCmds property in the [WebSpeed.Messengers.CGIIP] section.

NOTE: Do change the AllowsMsngrCmds property in the [WebSpeed.Messengers] section.Changes in this section will not affect the AllowsMsngrCmds property for individual messengers.

2 ♦ If the AllowsMsngrCmds property is disabled, set the property value to 1 to enable it.

3 ♦ Add the wsmAdmIPList property and type each unique IP address that you want to identify, separating each address with a comma.

CAUTION: Do not leave the IP address field blank (default value) because it will enable any IP address to access the WebSpeed Messenger's internal commands.

A.1.4 Check the Status Of the Debug Mode

The Debug mode allows you to examine and correct the errors that display in the browser when you are running code. This mode can provide information about the WebSpeed Agent, the system, and the path of an application.

When your application is in Production mode, the Debug mode is always off, regardless of the value on the Debug setting. When you are in Development mode, you can set values for the Debug mode to enable or disable debugging. Table A–3 and Table A–4 detail how each of these modes govern the values associated with the Debug mode.

Procedure Using the Progress Explorer Tool

To check the status of the Debug mode using the Progress Explorer Tool, perform these steps:

1 ♦ Double click the WebSpeed folder on the left-hand side of the main Progress Explorer window.

2 ♦ Right click the broker object.

3 ♦ Choose Properties from the shortcut menu.

4 ♦ Click on Agent→ Advanced Features to display properties associated with the WebSpeed Agent. Figure A–5 shows the Debug mode field and its three options.

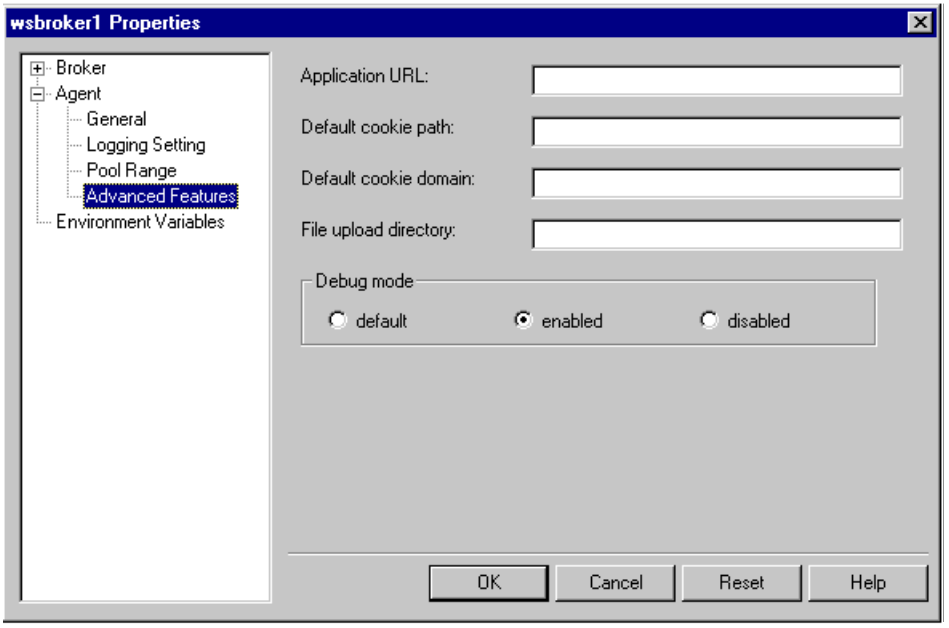

Figure A–5: Debug Mode Field

5 ♦ The Agent application mode value governs which Debug mode values are available:

– If the value set for the Agent application mode is Production, the Debug mode is always off, regardless of the value that displayed on the Debug setting.

– If the value set for the Agent application mode is Development, you can set values for the Debug mode that allow you to enable or disable debugging.

Table A–3 and Table A–4 detail how each mode governs the values associated with the Debug mode.

Procedure Using the ubroker.properties File

To check the status of Debug mode from the `ubroker.properties` file, perform these steps:

1 ♦ Review the SrvrAppMode setting for the broker property sections that you want to affect. It will be either Production or Development.

The specific mode you are in governs how the system interprets the value you set for the Debug mode. Table A–3 and Table A–4 detail this information.

2 ♦ Add the srvrDebug property to the broker property sections that you need to change.

3 ♦ Set the value for the srvrDebug mode.

Table A–3 describes in detail how the Production mode governs the Debug mode. Similarly, Table A–4 details how the Development mode governs the Debug mode.

Table A–3: How the Production Mode Effects the Debug Mode Values

When the Debug mode is...	Then...
[blank]	The Debug mode displays either the enabled or disabled value set at the srvrAppMode level. However, regardless of the value set at the srvrAppMode and subsequently displayed as the default value for the Debug mode, the Debug mode is off.
Enabled	The Debug mode is automatically off.
Disabled	The Debug mode is automatically off.

Table A–4: How the Development Mode Effects the Debug Mode Values

When the Debug setting is...	Then...
[blank]	The Debug mode displays and honors the default value set at the srvrAppMode level. That is, if the default value is defined as enabled, then the Debug mode is on. If the default value is defined as disabled, the default value is off.
Enabled	The Debug mode is set to on; you can debug files.
Disabled	The Debug mode is set to off; you cannot debug files. **NOTE:** If the Debug mode is set to Disabled, then specifying `?debug=on` in the URL has no effect.

A.2 Changing Additional Settings To Minimize Security Risks

The following information identifies some additional changes you can make when you change your application from a Development to a Production mode. These changes further limit your system's exposure to unauthorized access.

A.2.1 Change Default Port and Default WebSpeed Broker/Server Name Values

Assign random port numbers and change otherwise predictable WebSpeed Broker and Server Names to help mask known, default information. These changes makes it more difficult for personnel outside your organization who know the Progress WebSpeed product to access your files.

- Change all default port numbers to random, available port values between the range of 1025 and 65536 before you deploy WebSpeed. Table A–5 highlights some key default port numbers to consider changing.

Table A–5: Typical Default Port Numbers To Change

This default port number...	Runs...
20931	The Admin Server.
5162	The Name Server.
3050	Any new WebSpeed Brokers.
3055	The predefined wsbroker 1 WebSpeed Broker.

- Change all default WebSpeed Broker names, Admin Server names, and Name Server Names from their system-supplied default names to proprietary names to protect their identities.

A.2.2 Minimize the PROPATH Entries

Restrict the entries in the PROPATH to application and DLC/tty files; this technique allows you to expose only the information that is required to run WebSpeed applications.

Do not include references to the following information in the PROPATH:

- FTP directories
- Any File Upload directory set in your `ubroker.properties` file
- Any Web server directories
- Progress-supplied procedure libraries

A.2.3 Techniques To Minimize Unauthorized Access To WebSpeed Messengers

Consider replacing the default filenames and file extensions associated with key WebSpeed files with proprietary (that is, less easily recognizable) names and extensions that you define. This action limits unauthorized personnel's ability to recognize these files when they display on the URL.

The following list identifies these key files and suggests some ways to shield the identify of each file:

- Rename the default WebSpeed Messenger filename associated with the messenger type that you are using: cgiip.exe, wsisa.dll, or wsnsa.dll.
- Use a file association technique to shield the identity of the default WebSpeed Messenger and Broker filenames when they are run. This activity is only supported if you are using a Microsoft Internet Information Web Server (ISS Web Server) on a Windows NT platform, and your WebSpeed Messenger type is cgiip.exe. This technique allows you to define a file extension that can run an executable. The file extension, which includes the default filenames of the WebSpeed Messenger and Broker, obscures the identity of these files as it passes the Broker name to an executable that runs them.

  For detailed instructions on how to perform this file association technique, refer to the cgiip.wsc file that is shipped with the WebSpeed product.
- If you are using a UNIX platform, consider changing the default script name, `wspd_cgi.sh`, to a less immediately identifiable name to hide the WebSpeed messenger and WebSpeed Broker names that the `wspd_cgi.sh` file contains.

Restrict Your File Upload Directory

Do not allow execute permissions on your file upload directory.

NOTE: As previously mentioned earlier in this section, do not include references to the file upload directory in your PROPATH.

WebSpeed Configuration Considerations

As a general guideline to restrict access to your WebSpeed configuration from attack, protect your WebSpeed Broker, Web server, and database components inside your firewall; your Web server and WebSpeed messenger must be on a machine outside this firewall.

NOTE: For a more information about security issues as they pertain to WebSpeed configurations and their integration with firewalls, see the section, "Maximizing WebSpeed Compatibility With Your Firewall".

A.3 Authenticating a Password Using SpeedScript

As part of the basic, essential security measures for your site, it is recommended that you

- Ensure that a user is, in fact, a valid user who is recognized by the system
- Grant this authenticated user access only to the system resources to which this individual is assigned, or authorized

These measures are typically referred to as *authentication* and *authorization*, respectively. There are several commonly accepted methods to implement authentication and authorization, such as authentication based on username and password (for which a SpeedScript-based example is provided in this section), standard strong authentication, and Web browser to Web server authentication.

A.3.1 Validating a Password Using SpeedScript

This section describes how to validate a password in WebSpeed by executing a simple authentication procedure based on username and password. In the procedure, authorization and password variables are passed using an environment variable as written in SpeedScript. To demonstrate the steps in this authentication process, this example is designed to validate the hard-coded value mypass. The Post form method is used to protect the value of your password selection.

NOTE: The Get method is not recommended in this type of authentication process because it automatically displays the value of your password in your browser's URL field.

You can also modify this example to query a table that contains user and password information that must be verified.

1 ♦ Create a blank HTML file.

2 ♦ Enter the following code:

```
<html>
<head>
<title>Password Example</title>
</head>
<body>
<form method="post" action="val.html"
Enter Password:<input type="password" name="pass" value=""><br>
<input type="submit" name="action" value="process">
</form>
</body>
</html>
```

3 ♦ Close and save the file.

4 ♦ Create another blank HTML file and enter the following code:

```
<html>
<head>
<title>Password Page 2</title>
</head>
<body>
<script language="SpeedScript">
define variable pass as character.
If get-value("pass") = "mypass" then do:
{&out} "Accepted".
/*Could also run an .html or .r file from here, using the run statement.*/
end.
else do:
{&out} "Denied".
/*Could also redirect the user to an .html or .r file from here, using
the run statement.*/
end.
</script>
</body>
</html>
```

5 ♦ Close and save this second file with the filename *val.html*.

6 ♦ Run the first HTML file you created in Step 2.

The word **Accepted** displays to verify the acceptance and validation of the username and password values.

A.4 Securing Data Transmissions Over the Internet

Transmitting data over the Internet exposes your WebSpeed site to a host of potentially hostile activities. Using the Secure HTTP Protocol (HTTPS) and the Secure Sockets Layer (SSL) you can transmit data to and from external destinations in a secure manner. This section briefly discusses how your WebSpeed applications can benefit from each of these industry-accepted protocols.

A.4.1 Using Secure HTTP Protocol (HTTPS)

The Secure Hypertext Transport Protocol, also referred to as HTTPS, is a security-enhanced extension of the Hypertext Transmission Protocol (HTTP). The HTTPS is designed to support transmission of individual messages, or Web pages, securely over the Internet.

WebSpeed is compatible with HTTPS. HTTPS adds application-level encryption and security on top of ordinary sockets-based communications. The client and server communicate over any ordinary HTTP session and then negotiate their security requirements. They can use public and private keys to encrypt and de-encrypt messages transmitted using HTTPS.

HTTPS provides many benefits for WebSpeed users to support their e-commerce efforts; encryption, authentication, and message integrity are just a few of the benefits of using this protocol.

A.4.2 Using Secure Sockets Layer (SSL)

SSL has been widely accepted on the Internet as the preferred, secure protocol to authenticate and encrypt communication between clients and servers. Where HTTPS only secures individual messages, SSL secures the points of origin and destination for any amount of data that can be transmitted between a client and server. This protocol standard further complements and extends the security measures available to protect the confidentiality of your communications.

The SSL protocol resides above the network protocol, as defined by Transmission Control Protocol/Internet Protocol (TCP/IP), and directly below the application protocols, such as HTTP, HTTPS, or IMAP. It uses TCP/IP on behalf of the application protocols and, in the process, enables the following activities to occur:

- An SSL-enabled server can authenticate itself to an SSL-enabled client
- An SSL-enabled client can authenticate itself to an SSL-enabled server
- The SSL-enabled server and client can establish an encrypted connection

A.5 Maximizing WebSpeed Compatibility With Your Firewall

This section reviews ways to enhance your site interaction with your firewall.

The main purpose of a firewall is to restrict access to your Web site to ensure that only authorized, authentic users are allowed to connect through specific communication access, or controlled, points. The following topics highlight important security issues related to Progress and WebSpeed communication ports and traffic that interact with the firewall:

- WebSpeed TCP Port Details
- Important WebSpeed and Firewall Considerations
- Network Address Translation (NAT) and the WebSpeed Distributed Configuration
- Using the NameServer Client Port Range Value Settings With a Firewall

A.5.1 WebSpeed TCP Port Details

Your WebSpeed configuration might have specific firewall configuration requirements, especially when the WebSpeed Messenger is on a separate machine from the WebSpeed Broker. If a firewall exists between any two WebSpeed components, the firewall must be opened on the appropriate port.

Table A–6 identifies the network communications requirements for your WebSpeed configuration.

Table A–6: Network Communications Requirements *(1 of 2)*

Network Communication Links	Basic Requirements
Web browser to Web server	The Web browser must have access to the Web server. Typically, port 80 is used for this purpose; however, you can set up any port. **NOTE:** Your Web server configuration requires this connection.
WebSpeed Messenger to Name Server	The WebSpeed Messenger communicates with the Name Server using the User Datagram Protocol (UDP) that is open inbound on port 5162, or the Name Server port number. UDP ports also must be open outbound from the Name Server to the WebSpeed Messenger to complete the communication process.

Table A–6: Network Communications Requirements *(2 of 2)*

Network Communication Links	Basic Requirements
WebSpeed Messenger to WebSpeed Broker	The WebSpeed Messenger communicates with the WebSpeed Broker on a specific port that is defined when the WebSpeed Broker is configured. You can check this information either in the Progress Explorer or in the `install-dir/properties/ubroker.properties` file.
WebSpeed Messenger to WebSpeed Agents	The WebSpeed Messenger also communicates directly with WebSpeed Agents. When you configure the WebSpeed Broker, you must specify a port range for the agents to use. You can check this information either in the Progress Explorer or in the `install-dir/properties/ubroker.properties` file.
WebSpeed Agents to database	WebSpeed Agents are Progress clients. Use the Progress client-server configuration to establish network communication between these clients and their associated database.

A.5.2 Important WebSpeed and Firewall Considerations

The following information highlights key considerations regarding WebSpeed and firewall configuration:

- It is not advisable to set up a firewall between WebSpeed and a remote database. This type of arrangement can unnecessarily expose your site to security leaks on many levels. It is preferable that you set up your WebSpeed Brokers and Agents with your database behind the same firewall.
- Avoid opening the database directly to the Internet.

A.5.3 Network Address Translation (NAT) and the WebSpeed Distributed Configuration

Network Address Translation, or NAT, is an Internet standard that supports one set of Internet Protocol (IP) addresses for external traffic and a second set of addresses for internal use. A router translates these IP addresses between network domains, hiding internal addresses and generating globally unique, routable addresses. NAT, and IP Convert, are general forms to mask IP addresses, shielding this information from the Internet. In this manner, the NAT process helps promote more secure connections between clients and WebSpeed.

Starting with Progress Version 9.1A and WebSpeed 3.1A, Progress and WebSpeed support NAT in a distributed configuration arrangement through a registration process. The value you select for the Registration Mode indicates how the WebSpeed Broker identifies its hostname when the WebSpeed Broker registers with its controlling NameServer. When a client application attempts to connect to an application service on a host machine that the WebSpeed Broker supports, the hostname information is passed to the client application, provided that the information can be authenticated.

The specific value you set up for the Registration Mode depends on the connection arrangement you want. The "Procedure Using the Progress Explorer" and the "Procedure Using the ubroker.properties File" identify and describe the Registration mode values.

Procedure Using the Progress Explorer

To make a client connection through a NAT firewall using the Progress Explorer Tool, perform these steps:

1 ♦ Double click the WebSpeed folder on the left-hand side of the main Progress Explorer window.

2 ♦ Right click the broker object.

3 ♦ Choose Properties from the shortcut menu.

4 ♦ Click on Broker→ Controlling NameServer Information to display properties associated with the WebSpeed Broker. Figure A–6 shows the location of the Registration Mode field and its three possible values.

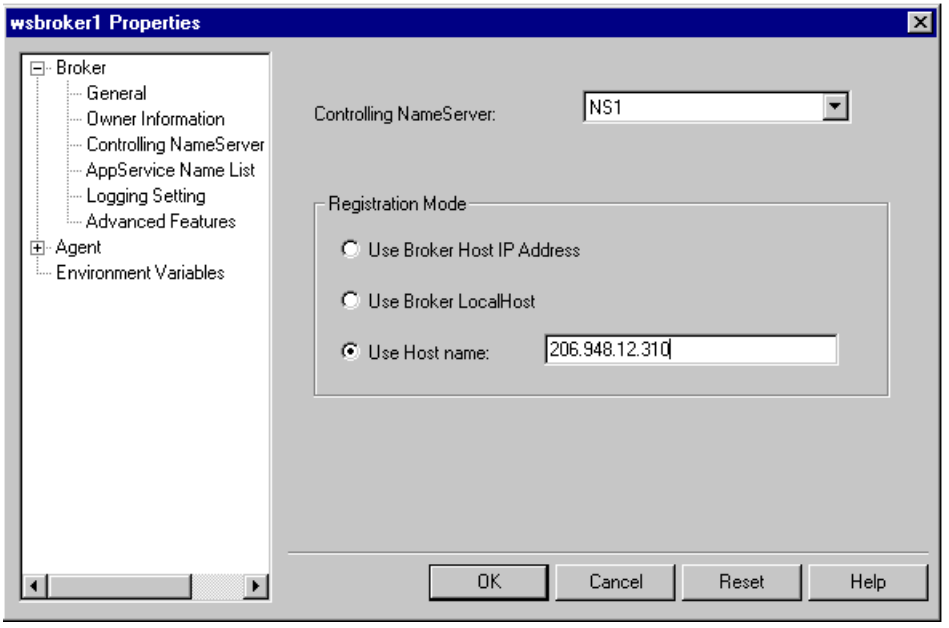

Figure A–6: Registration Mode Field

5 ♦ Select the Registration Mode value you want to set. Table A–7 defines each value.

Table A–7: Registration Mode Connection - Progress Explorer

Select this Registration Mode...	To Identify the hostname as...
Use Broker Host IP Address (default)	The IP address of the machine where the WebSpeed Broker is located. The WebSpeed Broker determines its IP address and passes this information to the NameServer when the Broker registers. Clients connect to the host using the Broker's IP address.
Use Broker LocalHost	The hostname of the machine where the WebSpeed Broker is located. The WebSpeed Broker determines its hostname and passes this information to the NameServer when the Broker registers. Clients connect to the host using this hostname.
Use Host Name (as defined in the unlabeled property description field)	The IP address defined. The WebSpeed Broker passes the IP address value defined in the property description field to the NameServer when the Broker registers. Clients connect to the host using the IP address defined.

6 ♦ Click the OK button.

Procedure Using the ubroker.properties File

To make a client connection through a NAT firewall using the `ubroker.properties` file, perform these steps:

1 ♦ Search the file for the broker property section, or sections, that you want to change.

2 ♦ Set the value of the registrationMode property. Table A–8 defines the values from which you can choose to enter in the registrationMode property.

Table A–8: RegistrationMode Connection - ubroker.properties File

Enter this Registration Mode...	To Identify the hostname as...
Register-IP (default)	The IP address of the machine where the WebSpeed Broker is located. The WebSpeed Broker determines its IP address and passes this information to the NameServer when the Broker registers. Clients connect to the host using the Broker's IP address.
Register-LocalHost	The hostname of the machine where the WebSpeed Broker is located. The WebSpeed Broker determines its hostname and passes this information to the NameServer when the Broker registers. Clients connect to the host using this hostname.
Register-HostName	A specific a hostname or IP address. You must • enter the value Register-HostName in the registrationMode property field • enter the hostname, or the IP address of the hostname, in the hostName property field to identify where the WebSpeed Broker is running • enter a description of the type of IP address you entered in the description property field The WebSpeed Broker passes either the hostname or the IP address value defined in the hostName property to the NameServer when the Broker registers. Clients connect to the host using either the hostName or the IP address defined.

NOTE: For more information, see the "GENERAL INSTRUCTIONS for configuring the Unified Broker and NameServer for WebSpeed and AppServers" in the `ubroker.properties` file.

A.5.4 Using the NameServer Client Port Range Value Settings With a Firewall

In Progress Version 9.1B, Progress introduced support for setting the NameServer client port range minimum and maximum value. The purpose of this section is to review this feature, and to describe how you can use it when there is a firewall between the WebSpeed Messenger and the NameServer.

The Progress WebSpeed Messenger communicates with the NameServer using the User Datagram Protocol (UDP). When the WebSpeed Messenger initiates the connection, it sends a UDP packet to the known NameServer port. When the NameServer receives the packet, it sends a response packet, also in UDP packet form, back to the WebSpeed Messenger.

To facilitate communication when a firewall exists between the WebSpeed Messenger and the NameServer, you could open all UDP ports from the machine that is running the NameServer to the machine that is running the WebSpeed Messenger.

However, opening all 65,000 UDP ports from inside the firewall to outbound ports is a time-consuming job. Also, opening every port is not a necessity. In Progress Version 9.1B, Progress introduced two new parameters that allow you to specify a range of UDP ports. You can specify a client port range minimum with `minNSClientPort and a` client port range maximum with `maxNSClientPort.`

Using these two new parameters, a firewall administrator can restrict the UDP response from the NameServer to the client. The administrator can specify a range of ports in the properties file and therefore reduce the number of UDP ports that are open in the firewall.

The following rules apply to the values set for these parameters:

- The value for these two parameters must be a number between 1024 and 65535 inclusive (or 0)
- The `minNSClientPort` also must be less than `maxNSCClientPort`

A.6 Additional Security Information Available On the Internet

Table A–9 highlights four Progress white papers. Each document presents detailed information about a general security topic, and specific, in- depth considerations as the topic relates to WebSpeed.

Table A–9: Supplemental Security-related Progress White Papers

The white paper...	Can be accessed by visiting...
Progress WebSpeed Strategy for Secure Implementation	http://www.progress.com/webspeed/whitepapers/strategy.htm
Securing Database Transactions on the Internet	http://www.progress.com/webspeed/whitepapers/securing_database.hts
Securing Your WebSpeed Application with an Authentication and Authorization Product	http://www.progress.com/webspeed/whitepapers/securing_authentication.htm
Securing Your WebSpeed Application with an Internet Firewall	http://www.progress.com/webspeed/whitepapers/securing_firewall.htm

B

State Management Using Connection Identifiers

This appendix is a supplement to the information on state management that is contained in the *WebSpeed Developer's Guide*. It describes how you can use unique connection identifiers to maintain context between web requests. Contents include:

- The Advantages Of Using SERVER-CONNECTION-ID
- Enabling Connection Identifiers

B.1 The Advantages Of Using SERVER-CONNECTION-ID

Using the SERVER-CONNECTION-ID attribute allows you to create a virtual session for your WebSpeed applications. You can maintain context between Web requests without tying up a WebSpeed Agent.

As described in the *WebSpeed Developer's Guide*, Web requests can be sent in a stateless mode or in a state-aware mode. In the stateless mode, web requests are independent of each other. No application context is maintained from one request to another. In state-aware mode, context is maintained by locking a WebSpeed agent to a particular client. State-aware mode has a scalability disadvantage because the locked agent is not available until a session is completed.

When you enable Session Connection ID, the WebSpeed Messenger checks incoming requests for a cookie containing a value for SERVER-CONNECTION-ID. The SERVER-CONNECTION-ID variable contains a unique identifier. If it does not exist, the WebSpeed Messenger causes a unique identifier to be generated. The WebSpeed Messenger then sets the SERVER-CONNECTION-ID attribute of the SESSION handle to the value of the unique identifier. (For more information, see the *Progress Language Reference*.) This unique connection identifier gets passed to the WebSpeed Agent as part of the Web request. In turn, the WebSpeed Agent passes the unique connection identifier to the WebSpeed application (also known as the Web object).

It is important to be aware that your Web object **will not** automatically associate the current request with a prior request even though it receives a unique connection identifier. You will need to create a data source where you can save connection identifiers and their associated application context. Then, you must create the logic in the Web object that:

- checks for SESSION: SERVER-CONNECTION-ID
- determines if the connection identifier is new or if it already exists in your data source
- creates a new entry in your data source if the connection identifier is new
- restores application context if the connection identifier already exists in your data source
- saves the changes in application context with the appropriate connection identifier before it completes execution

B.2 Enabling Connection Identifiers

The WebSpeed Messenger must be enabled to check for connection identifiers and to create them if they do not already exist.

The easiest way to do this is to modify the WebSpeed Messenger properties in Progress Explorer. As shown in Figure B–1, right click on the appropriate Messenger name and then select Properties. The Messenger's properties dialog will appear.

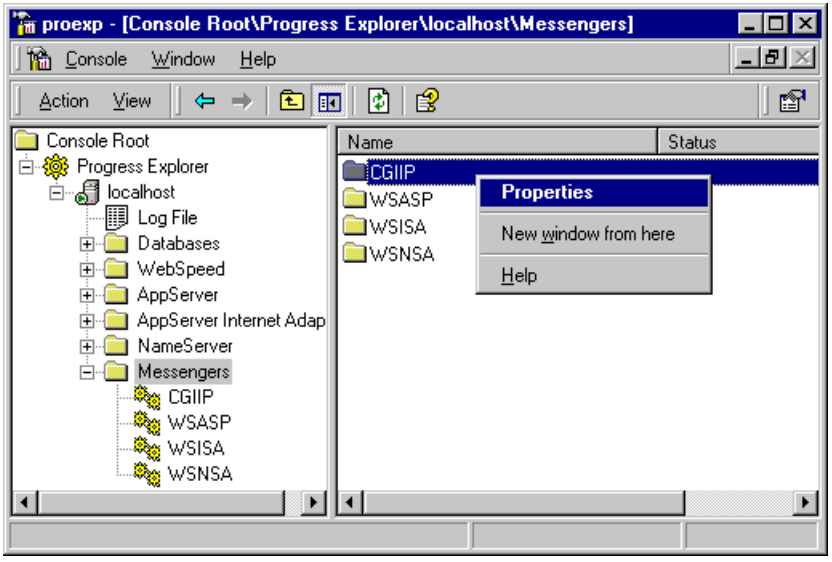

Figure B–1: Using Progress Explorer To Change Messenger Properties

In the Messenger properties, choose the Session Connection ID button, as shown in Figure B–2. Then choose OK. Restart the Web Server if the Messenger is WSASP, WSISA, or WSNSA. You do not have to restart the Web Server for CGIIP Messengers

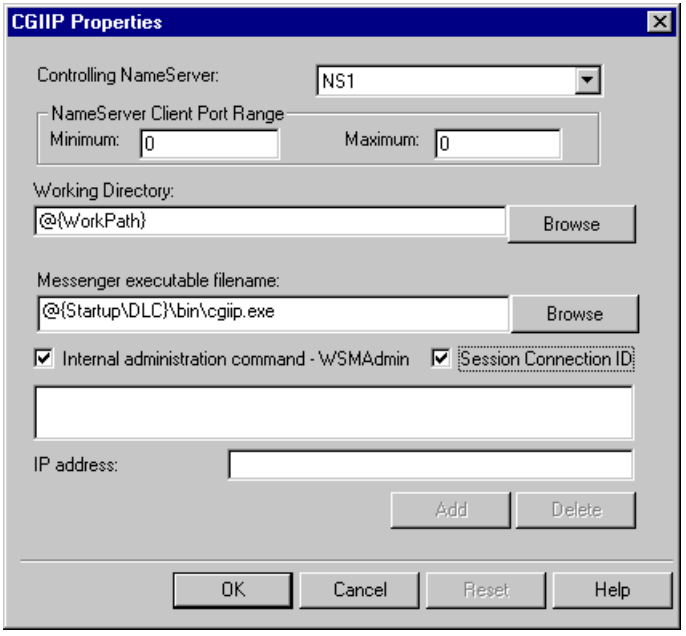

Figure B–2: The WebSpeed Messenger Properties Dialog Box

C

Using Active Server Pages With WebSpeed

This appendix contains information for developers who want to use Microsoft's Active Server Page (ASP) technology in conjunction with their WebSpeed applications. Contents include:

- Setting Up the WebSpeed ASP (WSASP) Messenger
- Accessing the WebSpeed ASP Web Page
- Enabling the WebSpeed ASP Example

For more information about ASP technology, go to:

```
http: //msdn.microsoft.com/workshop/server/asp/ASPover.asp
```

C.1 Setting Up the WebSpeed ASP (WSASP) Messenger

The WebSpeed Active Server Page (WSASP) Messenger allows you to integrate WebSpeed applications with Microsoft Active Server Pages. This integration allows you to use Microsoft's server side scripting and ActiveX server components.

In order to enable the use of Active Server Pages, you must set up the WSASP Messenger on a machine that has WebSpeed installed and is running a Microsoft Web server (for example, IIS version 3.*x* or 4.*x*). To set up the WSASP Messenger, do the following:

1 ♦ Log in as administrator on the machine that is running the Web server.

2 ♦ Shut down the Web server.

3 ♦ Register the `wsasp.dll` file:

a) Open a DOS window.

b) Change directory to `$DLC/bin`.

c) Enter: `regsvr32.exe wsasp.dll`

4 ♦ Restart the Web server.

5 ♦ Test your configuration.

You can test your configuration (and also customize error messages) from the WebSpeed Configuration and Verification Page. A link to the WebSpeed Configuration and Verification Page is on the WebSpeed ASP Web Page. See "Accessing the WebSpeed ASP Web Page" for more information.

C.2 Accessing the WebSpeed ASP Web Page

The WebSpeed ASP Web page allows you to:

- test the WSASP Messenger
- customize error messages
- view documentation on the properties and methods of the WSASP messenger
- run a sample application

To access the WebSpeed ASP Web page, you must create a virtual directory called `wsasp` from the Microsoft Web server console. The virtual directory must point to `$DLC/webspeed/wsasp` and must have execute permissions.

When you go to `http://`*`host-name`*`/wsasp`, you should see the page shown in Figure C–1.



Figure C–1: The WebSpeed ASP Web Page

Before you attempt to run the sample ASP application, see "Enabling the WebSpeed ASP Example."

C.3 Enabling the WebSpeed ASP Example

Before you can run the ASP example, you must do the following:

1 ♦ Create a working copy of the Sports2000 sample database.

2 ♦ Start a server for the working copy of the Sports2000 database.

3 ♦ Start a WebSpeed broker named wsbroker1 whose agents are connected to the Sports2000 database server.

NOTE: See *Getting Started with WebSpeed* for more detailed instructions on how to copy, start, and connect to the Sports2000 database.

4 ♦ Make wsbroker1 the default service.

You can do this in Progress Explorer by selecting AppService Name List from the wsbroker1 Properties dialog box. Make sure that the Supports default service check box is selected as shown in Figure C–2.

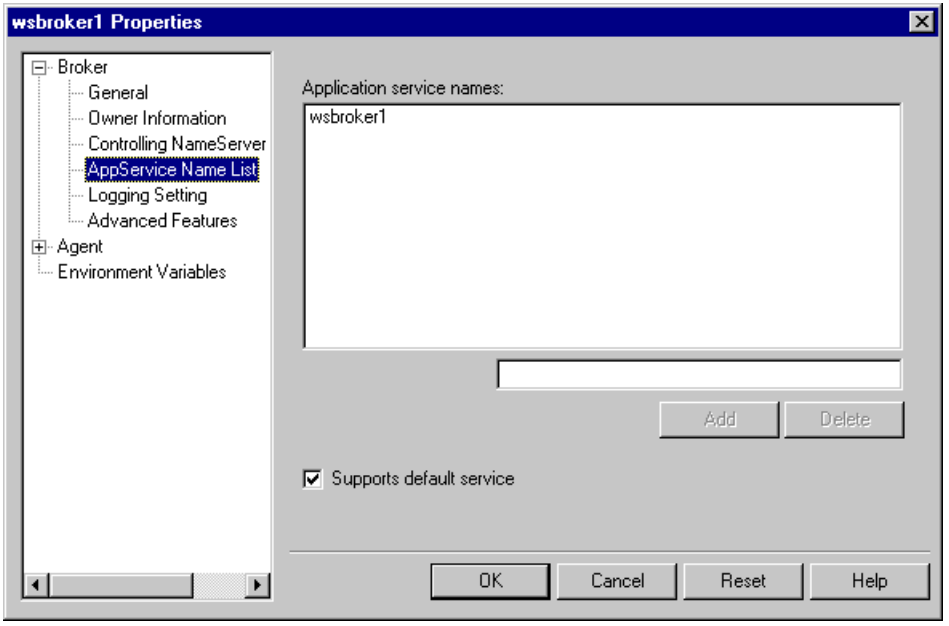

Figure C–2: wsbroker1 Properties Dialog Box

D

Using JavaScript With WebSpeed

This appendix contains information for developers who want incorporate JavaScript into their WebSpeed applications. Contents include:

- SpeedScript Versus JavaScript
- Using JavaScript Source Files
- Some JavaScript Examples

D.1 SpeedScript Versus JavaScript

It is a common practice to use both SpeedScript and JavaScript when developing WebSpeed applications. SpeedScript has advantages for developing the business logic of a applications, while JavaScript is a good programming tool for adding user interface elements to Web applications.

If you use either the Report or Detail Wizards in AppBuilder to create a WebSpeed Web object, you can view the resulting HTML source file and see a combination of SpeedScript and JavaScript. The wizards will create SpeedScript to implement database queries and updates, and they will create JavaScript event handlers (like onMouseOver, onClick, etc.,) to implement interactive features of the WebSpeed applications.

The <SCRIPT> tag for JavaScript employs the same syntax as the <SCRIPT> tag for Embedded SpeedScript:

```
<SCRIPT Language="JavaScript">
JavaScript code
</SCRIPT>
```

In some situations, you do not need a <SCRIPT> tag. JavaScript event handlers, for example, do not require a <SCRIPT> tag when they are used as an attribute to an HTML tag, as shown in the following:

```
<BODY onLoad="alert('Done');">
```

Some other factors that you should keep in mind when using JavaScript in WebSpeed applications are:

- End users of your WebSpeed application will be able to see your JavaScript code when they view HTML source in their browsers. They can see the HTML output that Embedded SpeedScript generates, but they do not see the actual SpeedScript source code. (This is because the SpeedScript code executes on the server side while the JavaScript executes on the client-side browser.)
- No static or dynamic HTML can be generated from the JavaScript code that is between HTML <SCRIPT> tags.
- SpeedScript is executed on the server side by the WebSpeed Agent. JavaScript is executed on the client side by the Web browser.

D.2 Using JavaScript Source Files

If you develop JavaScript routines that are long and that are common to a number of applications, you may want to keep them in separate JavaScript source (`.js`) files.

When deploying JavaScript source files, you must put them in the Web server's root directory or in a subdirectory of the Web server's root directory. You can then reference the file using the HTML SRC attribute. For example, the following calls `myscript.js`, which is in `jsscripts` a subdirectory of the Web server's root directory:

```
<SCRIPT LANGUAGE="JavaScript" SRC="/jsscripts/myscript.js">
```

You can also create references to JavaScript source files within Embedded SpeedScript. However, you must be careful to specify the complete `<SCRIPT>` tag expression and to escape the expression using back tics. For example, the following code snippet **will not** run the JavaScript file:

```
<SCRIPT LANGUAGE="SpeedScript">
DEFINE VARIABLE myVar AS CHARACTER NO-UNDO.
myVar = "/jsscripts/myscript.js".
</SCRIPT>

<H3>Trying to run JavaScript ...</H3>
<SCRIPT LANGUAGE="JavaScript" SRC="`myVar`">
</SCRIPT>

<H3>JavaScript did not run...</H3>
```

The problem is fixed when you assign the complete `<SCRIPT>` tag expression to the variable and enclose it in back tics:

```
<SCRIPT LANGUAGE="SpeedScript">
DEFINE VARIABLE myVar AS CHARACTER NO-UNDO.
myVar='<SCRIPT LANGUAGE="JavaScript"
SRC="/jsscripts/myscript.js"></SCRIPT>'.
</SCRIPT>

<H3>Now running JavaScript ...</H3>
`myvar`

<H3>JavaScript done ...</H3>
```

D.3 Some JavaScript Examples

It is beyond the scope of this book to give you detailed instructions on using JavaScript. However, this section gives a few examples that you might find useful in your WebSpeed applications.

D.3.1 A Combo Box With URL Buttons

This example shows you how to code a combo box to hold URLs. When the user clicks on the button, the browser goes to the Web site specified in the URL:

```
<FORM>
 <SELECT SIZE="1"
   onChange="(this[selectedIndex].value ?
              location=this[selectedIndex].value : null)">
   <OPTION SELECTED>Select Location
   <OPTION VALUE="http://www.progress.com">Progress
   <OPTION VALUE="http://www.progress.com/services/support">Tech Support
   <OPTION VALUE="http://www.webspeed.com">WebSpeed
 </SELECT>
</FORM>
```

D.3.2 Browser Detection

The following JavaScript code snippet contains a simple method for determining whether a browser is Internet Explorer or Netscape Navigator. It tests for the existence of the document.all object, which is only available on Internet Explorer. Then it tests for the existence of the document.layers object, which is only available on Netscape Navigator:

```
var isIE = false;
var isNav = false;
if (document.all)
  isIE = TRUE;
else if (document.layers)
  isNav = TRUE;
```

An alternative method for browser detection is to use SpeedScript to create a test in your WebSpeed web object which, of course, runs on the server side. The following test, written in SpeedScript, checks for the value of the CGI environment variable, HTTP_USER_AGENT:

```
IF INDEX(get-cgi('HTTP_USER_AGENT':U)," MSIE ":U) > 0 THEN isIE = TRUE.
```

CGI environment variables are set automatically when web requests are received by a WebSpeed agent. In this case, if the value of HTTP_USER_AGENT is MSIE, then the client request came from an Internet Explorer Web browser.

D.3.3 Printing From an HTML Page

This example shows how a Web page can print itself after it finishes loading in the Web browser:

```
<BODY onLoad="window.print()">
This example will pop up the printer dialog box when it's loaded into the
browser. The intention of this example is for WebSpeed users to create
applications that can print HTML documents without browser interaction.
</BODY>
```

The next example shows how you can print a portion of a page dynamically. First, you must set up two frames, one that takes up the entire browser window and another that is hidden:

```
/* startup.w */
<FRAMESET FRAMEBORDER=1 COLS="100%,*">
  <FRAME ID="appFrame" SRC="myApp.w" SCROLLING="no">
  <FRAME ID="jsFrame" SRC="empty.html" SCROLLING="no"
  NORESIZE FRAMEBORDER=0 MARGINHEIGHT=0 MARGINWIDTH=0>
</FRAMESET>
```

The appFrame contains the application. The jsFrame is used for communicating with the server and for running applications like printing.

The following shows the filePrint() function in the simplified myApp.w code fragment:

```
/* myApp.w fragment */
<SCRIPT LANGUAGE="JavaScript">
function filePrint() {
var newPage = '<PRE>' + document.form1.elements["textarea0"].value;
  parent.jsFrame.document.write(newPage);
  parent.jsFrame.document.close();
  parent.jsFrame.focus();
  parent.jsFrame.print();
}
</SCRIPT>
```

The filePrint() function copies the value of a <TEXTAREA> field named textarea0 to the hidden frame with a <PRE> prefix. Focus is moved to that frame and then it is printed. If you eliminate the <PRE> prefix, the browser renders the context of textarea0 before it is printed.

E

WebSpeed Dynamic Code-page Support

This appendix describes how to incorporate dynamic code-page support into WebSpeed applications. This information is for an experienced WebSpeed developer familiar with:

- Code pages, code-page conversion, and code-page compatibility
- WebSpeed dynamic code-page support

For more information on code pages, see the *Progress Internationalization Guide*. For more information on WebSpeed dynamic code-page support, see the *WebSpeed Installation and Configuration Guide* and the reference entry for the HTML-CHARSET attribute in the *Progress Language Reference*.

This appendix contains the following sections:

- Overview
- Determining the MIME Code-page Name
- Storing the MIME Code-page Name
- Accessing WEB-CONTEXT's HTML-CHARSET Attribute
- Converting Code-page Names Between MIME and Progress Formats
- Additional Notes

E.1 Overview

To summarize dynamic code-page support from the perspective of the WebSpeed developer:

1. Progress Software Corporation recommends that the WebSpeed application includes a page that lets the user select a language or nationality from a list of preselected languages or nationalities. For each language or nationality in the list, the developer determines an appropriate code page, which must be compatible with the WebSpeed Agent's CPINTERNAL code page. To simplify meeting this requirement, make the Agent's CPINTERNAL code page UTF-8, which is highly compatible with Progress code pages. (For more information, see the `$DLC/prolang/readme` file.)

2. The user starts the application and selects a language or nationality.

3. The application notes the language or nationality selected, determines the corresponding code page, and stores the MIME version (as opposed to the Progress version) of the code-page name within the Web request in one of several places known to the Agent.

   NOTE: Progress Software Corporation recommends that once the user chooses the code page, any static HTML the application uses should be in the same code page, and should be labeled with the same code page in the `<META>` tag's `HTTP-EQUIV` attribute near the top of the HTTP document. For an example of one way to do this, see the first code example in the "Storing the MIME Code-page Name" section in this chapter.

4. Dynamic code-page support takes over.

   Each time the Agent receives information from a Web browser, Progress automatically converts the information from the browser's code page to the Agent's CPINTERNAL code page.

   Similarly, just before the Agent sends a Web page back to the browser, Progress automatically converts it from the Agent's CPINTERNAL code page to the browser's code page.

E.2 Determining the MIME Code-page Name

The application must store the MIME code-page name, not the Progress code-page name. Table E–1 gives the MIME name and the Progress name for several common code pages.

Table E–1: MIME and Progress Code-page Equivalents

MIME Name	Progress Name
iso-8859-2	iso8859-2
Shift_JIS	SHIFT-JIS
TIS-620	620-2533
windows-1250	1250

NOTE: Table E–1 does not list all code pages that can be used with dynamic code-page support. For a complete list of these code pages, see the source code for `adecomm/convcp.p`.

For more information on converting between MIME and Progress code-page names, see the "Converting Code-page Names Between MIME and Progress Formats" section.

E.3 Storing the MIME Code-page Name

To store the MIME code-page name, use one of the following techniques:

- Using HTML, store the MIME code-page name in a hidden form-field or a query field called wscharset.

  NOTE: The field must be called wscharset.

  In the following fragment, the MIME code-page name is stored in a hidden form-field called wscharset:

```
<HEAD>
<META HTTP-EQUIV="Content-Type"
  CONTENT="text/html; charset=windows-1252">
</HEAD>
<BODY>
<FORM METHOD="POST" ACTION="test.w">
  Name: <INPUT TYPE="text" ID="name"><BR>
  Address: <INPUT TYPE="text" ID="address"><BR>
  City: <INPUT TYPE="text" ID="city"><BR><BR>
  <INPUT TYPE="hidden" NAME="wscharset" VALUE="windows-1252">
  <INPUT TYPE="submit" VALUE="Submit">
</FORM>
</BODY>
```

  In the following fragment, the MIME code page is stored in a query field called wscharset:

```
<A HREF="tstiso2.w?wscharset=iso-8859-2"></A>
```

- In 4GL, use the SET-COOKIE API function to create a cookie called wscharset containing the MIME code-page name, as shown in the following fragment:

  NOTE: The cookie must be called wscharset.

```
SET-COOKIE("wscharset":U, "iso-8859-2":U, ?, ?, ?, ?, ?).
```

CAUTION: Dynamic code-page support does not work if the code-page name is stored using the CHARSET attribute of the META tag exclusively or the WebSpeed API outputContentType() function (which does the same) exclusively. You must also store the code-page name in wscharset, as described in this section.

CAUTION: If you store the code-page name using the CHARSET attribute of the META tag or the WebSpeed API outputContentType() function, and you store the code-page name in wscharset as described in this section, the stored code-page names must match.

E.4 Accessing WEB-CONTEXT's HTML-CHARSET Attribute

The WebSpeed Agent stores the Progress version (as opposed to the MIME version) of the code-page name in WEB-CONTEXT's HTML-CHARSET attribute. Once HTML-CHARSET is assigned this value, it can be accessed by the WebSpeed application.

NOTE: Progress Software Corporation recommends that the WebSpeed application **not** modify WEB-CONTEXT:HTML-CHARSET.

For complete information on HTML-CHARSET, see its reference entry in the *Progress Version 9 Product Update Bulletin* and *Pocket WebSpeed*.

E.5 Converting Code-page Names Between MIME and Progress Formats

You can convert a code-page name between MIME and Progress formats by running the `convcp.p` procedure, which resides in the `adecomm` directory.

Here is the syntax:

SYNTAX

```
convcp.p(input-name, direction:U, OUTPUT output-name)
```

input-name

A CHARACTER expression indicating the code-page name to be converted.

direction

A CHARACTER indicating one of the following:

- "toProg" (to convert from MIME to Progress)
- "toMime" (to convert from Progress to MIME)

output-name

A CHARACTER expression indicating the name of a CHARACTER variable to contain the result.

If *input-name* is not a valid MIME or Progress code-page name, *output-name* is set to blank.

The following example converts the value of cMimeCharset to Progress format and stores the result in cProCharset:

```
RUN adecomm/convcp.p(cMimeCharset, "toProg":U, OUTPUT cProCharset).
```

The following example converts the value of cProCharset to MIME format and stores the result in cMimeCharset:

```
RUN adecomm/convcp.p(cProCharset, "toMime":U, OUTPUT cMimeCharset).
```

E.6 Additional Notes

Here are additional notes on developing applications with dynamic code-page support:

1. Progress Software Corporation recommends that the page for selecting a language or nationality not use fill-in fields. If fill-in fields are used, the application might not be able to determine which code page the fill-in fields are encoded in.

2. If your Web server can perform code-page conversion, turn this feature off if you want to use the dynamic code-page support supplied by WebSpeed.