

Getting Started with WebSpeed

© 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: 4418
Item Number: 89438;V3.1D

Contents

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

1. Welcome to the WebSpeed Workshop Environment **1–1**
1.1 The WebSpeed Workshop Environment 1–2
1.2 Supported Web Browsers 1–3
1.3 Supported Web Servers 1–3
1.4 Supported Authoring Tools 1–3

2. Introducing the AppBuilder and WebTools 2–1
2.1 The AppBuilder 2–2
2.1.1 Starting the AppBuilder 2–2
2.1.2 The AppBuilder Main Window 2–2
2.1.3 Visual and Nonvisual Objects 2–4
2.1.4 SmartDataObjects 2–4
2.1.5 Templates and Wizards 2–4
2.1.6 Editors 2–5
2.2 WebTools 2–6
2.2.1 Application Manager 2–7
2.2.2 Data Browser 2–7
2.2.3 Editor 2–7
2.2.4 File Tools 2–7
2.2.5 OS Command 2–8
2.2.6 Scripting Lab 2–8
2.2.7 Databases 2–8
2.2.8 Agent Variables 2–9
2.2.9 ProPath 2–9
2.2.10 Virtual System Tables 2–9
2.2.11 Object State 2–9
2.2.12 Messages 2–10
2.2.13 Home Pages 2–10
2.2.14 Online Help 2–11

3. Setting Up the WebSpeed Development Environment 3–1
3.1 Creating and Connecting To a Database 3–2
3.1.1 Making a Working Copy Of the Sports2000 Database 3–2
3.1.2 Starting a Server For the Database 3–4
3.1.3 Starting a WebSpeed Broker With Agents Connected To the Database 3–10
3.1.4 Connecting the AppBuilder To the Database 3–12
3.2 Setting AppBuilder Preferences For WebSpeed 3–14
3.2.1 Setting the Development Mode 3–17

4. Sample WebSpeed Applications 4–1
4.1 Running Sample Applications 4–2
4.2 Running SportsPro Sales Advisor 4–3
4.3 Source Code For Sample Applications 4–7

5. Creating a WebSpeed Application 5–1
5.1 Setting Up the WebSpeed Development Environment 5–2
5.2 Creating a Report With Search Capability 5–2
5.2.1 Starting the Report Wizard 5–3
5.2.2 Defining the Query 5–4
5.2.3 Selecting Fields For the Query 5–5
5.2.4 Defining the Style Of the Report 5–7
5.2.5 Previewing and Saving the Report 5–9
5.3 Creating a Master Report Linked To a Detail Update Form 5–10
5.3.1 Defining the Query 5–10
5.3.2 Selecting Fields For the Query 5–11
5.3.3 Defining the Style Of the Report 5–13
5.3.4 Defining the Links 5–14
5.3.5 Defining the Color Of the Report 5–15
5.3.6 Previewing and Saving Your Report 5–15
5.4 Using the Detail Wizard 5–16
5.4.1 Beginning 5–16
5.4.2 Defining the Query 5–16
5.4.3 Selecting the Fields For the Query 5–18
5.4.4 Defining the Style Of the Form 5–20
5.4.5 Defining the Links 5–21
5.4.6 Defining the Color Of the Form 5–22
5.4.7 Previewing and Saving Your Detail Update Form 5–23
5.5 Working With Templates 5–24
5.5.1 Creating a Welcome Screen 5–24
5.5.2 Saving Your Welcome Screen 5–24
5.5.3 Previewing Your Welcome Screen 5–25
5.5.4 Creating an Application Header 5–25
5.5.5 Saving Your Application Header 5–27
5.5.6 Previewing Your Application Header 5–27
5.5.7 Creating an Index 5–27
5.5.8 Saving Your Index 5–28
5.5.9 Previewing Your Index 5–29
5.5.10 Creating the Main Sales Advisor Frame Set 5–30
5.5.11 Saving Your Frame Set 5–30
5.5.12 Previewing Your Frame Set 5–31
5.6 Where To Go From Here 5–31

Glossary Glossary–1

Figures

Figure 2–1: The AppBuilder Main Window 2–2
Figure 2–2: WebSpeed WebTools Main Page 2–6
Figure 3–1: Dictionary Startup Dialog 3–2
Figure 3–2: Create Database Dialog 3–3
Figure 3–3: Connect Database Dialog 3–3
Figure 3–4: Progress Explorer Main View 3–5
Figure 3–5: Progress Explorer Expanded Tree View 3–6
Figure 3–6: New Object Dialog In Progress Explorer 3–7
Figure 3–7: Database Properties Dialog In Progress Explorer 3–7
Figure 3–8: Sports2000 In Progress Explorer 3–8
Figure 3–9: wsbroker1 In Progress Explorer 3–10
Figure 3–10: Broker and Agent Properties In Progress Explorer 3–11
Figure 3–11: Database Connections Dialog In AppBuilder 3–12
Figure 3–12: The Connect Database Dialog 3–13
Figure 3–13: WebSpeed Preferences Dialog 3–15
Figure 3–14: The Development Mode Button 3–17
Figure 4–1: Sample Applications Home Page 4–3
Figure 4–2: Sports Pro Sales Advisor Initial Frame Set 4–4
Figure 4–3: Inventory Report 4–5
Figure 4–4: Customer Report 4–6
Figure 4–5: Customer Update Form 4–6
Figure 5–1: The New Dialog 5–3
Figure 5–2: The Query Builder Dialog 5–4
Figure 5–3: Page 4 (Fields) Of the HTML Report Wizard 5–5
Figure 5–4: The Multi-Field Selector Dialog 5–6
Figure 5–5: Page 5 (Style) Of the HTML Report Wizard 5–7
Figure 5–6: Page 7 (Background) Of the HTML Report Wizard 5–8
Figure 5–7: Inventory Report 5–9
Figure 5–8: Page 2 (Data Source) Of the HTML Report Wizard 5–11
Figure 5–9: Column Editor 5–12
Figure 5–10: Page 5 (Style) Of the HTML Report Wizard 5–13
Figure 5–11: Page 6 (Hyperlink) Of the HTML Report Wizard 5–14
Figure 5–12: Customer Report 5–15
Figure 5–13: Query Builder 5–17
Figure 5–14: Page 4 (Fields) Of the Detail Wizard 5–19
Figure 5–15: Page 5 (Style) Of the Detail Wizard 5–20
Figure 5–16: Page 6 (Hyperlink) Of the HTML Detail Wizard 5–21
Figure 5–17: Page 7 (Background) Of the HTML Detail Wizard 5–22
Figure 5–18: Customer Update Form 5–23
Figure 5–19: Preview Of the Welcome Screen 5–25
Figure 5–20: Preview Of the Application Header 5–27
Figure 5–21: Preview Of the Index 5–29
Figure 5–22: Preview Of the Frame Set 5–31

Tables

Table 1–1: Browser Preferences and Settings 1–3
Table 2–1: The AppBuilder Menu Bar Options 2–3

Preface

Purpose

This book introduces you to the WebSpeed Workshop Environment, which is a suite of components and tools for creating Internet Transaction Processing (ITP) applications. It contains a brief description of the WebSpeed components and tools. It shows you how to set up WebSpeed to develop ITP applications. Finally, this book contains a tutorial that shows you how to use WebSpeed to create an ITP application.

Audience

This book is for anyone interested in learning how to create ITP applications with the WebSpeed Workshop Environment. Knowledge of WebSpeed or SpeedScript programming is not required. However, you should have some general experience with application development. You should also have a working understanding of the Internet and the World Wide Web, and some experience creating and editing HTML pages.

Organization Of This Manual

Chapter 1, “Welcome To the WebSpeed Workshop Environment”

This chapter introduces the WebSpeed Workshop Environment. The chapter also includes information about browser support and Web server support.

Chapter 2, “Introducing the AppBuilder and WebTools”

This chapter describes the AppBuilder, a Windows-based development tool. It also describes WebTools, which are browser-based development tools.

Chapter 3, “Setting Up the WebSpeed Development Environment”

This chapter describes how to set up WebSpeed prior to developing ITP applications. It focuses on the tasks you perform in order to run and develop the sample applications.

Chapter 4, “Sample WebSpeed Applications”

This chapter describes how to run the sample ITP applications that are included with WebSpeed. It includes a detailed description of one of the sample applications, the SportsPro Sales Advisor.

Chapter 5, “Creating a WebSpeed Application”

This chapter is a tutorial that describes how to create SportsPro Sales Advisor, a sample ITP application.

“Glossary”

The glossary provides useful definitions of WebSpeed and Internet terms.

How To Use This Manual

The information presented in this guide has been tailored to allow you to set up WebSpeed and complete the tutorial in Chapter 5 in less than half a day. However, you can complete as much of the tutorial as you want in any one session. To gain the most from the tutorial, complete the step-by-step instructions in Chapter 3 and in Chapter 5 in order.

Before you can run the sample applications or complete the tutorial, you must do a complete installation of WebSpeed Workshop or Progress ProVision Plus. For installation information, see the *WebSpeed Installation and Configuration Guide*.

Typographical Conventions

This manual uses the following typographical conventions:

- **Bold typeface** indicates:
  - Commands or characters that the user types
  - That a word carries particular weight or emphasis
- *Italic typeface* indicates:
  - Progress variable information that the user supplies
  - New terms
  - Titles of complete publications
- `Monospaced typeface` indicates:
  - Code examples
  - System output
  - Operating system filenames and pathnames

The following typographical conventions are used to represent keystrokes:

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

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

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

  CTRL-X

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

  ESCAPE H
  ESCAPE CURSOR-LEFT

WebSpeed Messages

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

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

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

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

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

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

Other Useful Documentation

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

Getting Started

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

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

Progress Installation and Configuration Guide Version 9 for UNIX

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

Progress Installation and Configuration Guide Version 9 for Windows

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

Progress Version 9 Product Update Bulletin

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

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

Progress Application Development Environment — Getting Started (Windows only)

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

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

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

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

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

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

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

Progress Startup Command and Parameter Reference

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

Welcome to Progress (Hard copy only)

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

Development Tools

Progress ADM 2 Guide

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

Progress ADM 2 Reference

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

Progress AppBuilder Developer's Guide (Windows only)

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

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

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

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

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

Progress Debugger Guide

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

Progress Help Development Guide (Windows only)

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

Progress Translation Manager Guide (Windows only)

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

Progress Visual Translator Guide (Windows only)

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

Reporting Tools

Progress Report Builder Deployment Guide (Windows only)

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

Progress Report Builder Tutorial (Windows only)

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

Progress Report Builder User's Guide (Windows only)

A guide for generating reports with the Progress Report Builder.

Progress Results Administration and Development Guide (Windows only)

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

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

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

4GL

Building Distributed Applications Using the Progress AppServer

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

Progress External Program Interfaces

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

Progress Internationalization Guide

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

Progress Language Reference

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

Progress on the Web

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

Progress Programming Handbook

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

Database

Progress Database Design Guide

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

Progress Database Administration Guide and Reference

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

DataServers

Progress DataServer Guides

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

Each DataServer has its own guide as follows:

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

MERANT ODBC Branded Driver Reference

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

SQL-89/Open Access

Progress Embedded SQL-89 Guide and Reference

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

Progress Open Client Developer's Guide

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

WebSpeed Installation and Configuration Guide

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

WebSpeed Developer's Guide

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

WebSpeed Product Update Bulletin

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

Welcome to WebSpeed (Hard copy only)

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

Reference

Pocket Progress (Hard copy only)

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

Pocket WebSpeed (Hard copy only)

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

1

Welcome To the WebSpeed Workshop Environment

WebSpeed Workshop is an environment in which you develop Web-based Internet Transaction Processing (ITP) applications: business applications that support real-time data access and data management.

This chapter introduces the WebSpeed Workshop environment and includes information about Web browser and Web server requirements. For more information about requirements, as well as installation and configuration information, see the *WebSpeed Installation and Configuration Guide*.

Contents of this chapter include:

- The WebSpeed Workshop Environment
- Supported Web Browsers
- Supported Web Servers
- Supported Authoring Tools

1.1 The WebSpeed Workshop Environment

The WebSpeed Workshop environment includes these components:

- **AppBuilder**—Depending on whether you purchased the WebSpeed product or the combined Progress and WebSpeed product, you might be using the AppBuilder for developing WebSpeed applications, Windows applications, or for both types of applications.

  The focus in this book is on using the AppBuilder as a WebSpeed development tool. For an introduction to the AppBuilder, see the *Progress Application Development Environment — Getting Started* guide.

- **WebTools**—You can use the browser-based WebTools to access the status of CGI variables, the state of objects, and a variety of database information. In addition, you can use the WebTools for creating, modifying, and managing files, for compiling and testing WebSpeed code, and for executing OS commands.

- **WebSpeed Development Server**—The WebSpeed Development Server is a development version of the WebSpeed Transaction Server. The WebSpeed Development Server allows you to run and test the applications you build with the AppBuilder. You also have access to the Procedure Editor for writing SpeedScript procedures; the Debugger, the Data Dictionary, and the Data Administration Tool for managing your development database; and PRO*Tools for managing your client development environment.

NOTE: The Color Changer, Screen Scaling Utility, and ProtoGen PRO*Tools do not apply to WebSpeed.

1.2 Supported Web Browsers

WebSpeed Workshop requires Netscape Navigator (Version 4.5 or later) or Microsoft Internet Explorer (Version 5.0 or later).

Table 1–1 lists the recommended Web browser settings for the WebSpeed Workshop environment.

Table 1–1: Browser Preferences and Settings

Preference	Setting
Disk Cache	5000kb
Memory Cache	1000kb
Number of Connections	4
Font	Small fonts, size 10 or 12 point
Java and JavaScript languages	Enabled
Temporary Internet Files (Internet Explorer only)	Check for newer versions of stored pages: Every visit to the page.

1.3 Supported Web Servers

For WebSpeed Version 3.1, you can use a Web server that supports one of the following interfaces:

- **CGI 1.1**—For example, Microsoft Internet Information Server (IIS), Versions 3.x and 4.x; Apache; Netscape Enterprise or Fast Track Server, Version 3.x
- **ISAPI**—For example, Microsoft Internet Information Server (IIS), Versions 3.x and 4.x
- **NSAPI**—For example, Netscape Enterprise or Fast Track Server, Version 3.x

1.4 Supported Authoring Tools

You can use an authoring tool, such as Microsoft FrontPage or Allaire Corporation's HomeSite, in conjunction with AppBuilder to create HTML and graphics files, or you can use the Progress Procedure Editor for coding HTML.

2

Introducing the AppBuilder and WebTools

This chapter introduces you to some of the tools and utilities used in WebSpeed application development. There are Windows-based and browser-based tools. You access the Windows-based tools through the AppBuilder. Browser-based tools, also known as WebTools, can be launched by starting a browser from the AppBuilder, or you can run them by supplying a URL directly in a Web browser.

Contents of this chapter include:

- The AppBuilder
- WebTools

2.1 The AppBuilder

The AppBuilder is a multipurpose application development environment. You can use it as a visual programming environment to create character- or GUI-based client/server applications. In addition, you can use the AppBuilder for WebSpeed to create HTML-based Web applications.

The AppBuilder is installed with one of the following licensing options:

- **WebSpeed AppBuilder**—supports creating, testing, and deploying Web objects
- **ProVision AppBuilder**—supports creating, testing, and deploying traditional n-tier client/server applications (either character- or GUI-based)
- **ProVision Plus AppBuilder** (**Unified AppBuilder**)—supports both client/server and Web application development, plus integration with other non-Progress applications such as Java or ActiveX

This book focuses on using the WebSpeed AppBuilder. For more information about the ProVision AppBuilder and ProVision Plus AppBuilder see the *Progress AppBuilder Developer's Guide*.

2.1.1 Starting the AppBuilder

Start the AppBuilder from within the Progress Program Group. For example, from the Start menu, choose Programs→ Progress→ AppBuilder. The main window of the AppBuilder opens.

In addition, you can start the AppBuilder from the Progress Application Development Environment (ADE) Desktop and any tool that has a Tools menu, like the Data Dictionary or the Procedure Editor. You can also start the AppBuilder directly from an operating system command shell. For more information about starting the AppBuilder, see the *Progress AppBuilder Developer's Guide*.

2.1.2 The AppBuilder Main Window

The main window (shown in Figure 2–1) contains a menu bar, a tool bar, fields that show information about the current object, and a status bar.

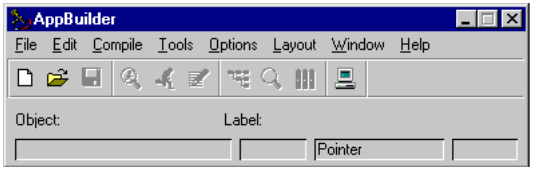


Figure 2–1: The AppBuilder Main Window

The AppBuilder menu bar consists of eight menus: File, Edit, Compile, Tools, Options, Layout, Window, and Help. Table 2–1 describes how these menus relate to WebSpeed application development.

Table 2–1: The AppBuilder Menu Bar Options

The options in this menu...	Allow you to...
File	Create a new Web object or open an existing Web object.
Edit	Edit the selected Web object.
Compile	Run, check the syntax of, and debug the application you create.
Tools	Access WebSpeed WebTools, manage database connections, and access other Progress tools and utilities.
Options	Connect to a WebSpeed Broker and specify your default browser.
Layout	Specify the location of visual objects (not available in WebSpeed AppBuilder or when creating Web objects).
Window	Manage multiple design windows.
Help	Access help topics and obtain information about messages you might receive when working with the AppBuilder.

See the *Progress AppBuilder Developer's Guide* for more information about the AppBuilder main window.

2.1.3 Visual and Nonvisual Objects

If you are using the Unified AppBuilder licensing option, a separate window called the Object Palette appears when you open the AppBuilder. If you are using the WebSpeed AppBuilder licensing option, the Object Palette does not appear. The Object Palette allows you to create visual objects (buttons, for example).

In WebSpeed, you only create nonvisual objects, which include embedded SpeedScript, CGI Wrapper, and HTML-mapping Web objects. (SpeedScript is an interpreted, block-structured, and statement-oriented language based on the Progress 4GL.) You cannot manipulate nonvisual objects graphically at design time, and they do not have a visualization element at run time.

2.1.4 SmartDataObjects

AppBuilder supports a nonvisual SmartObject called the SmartDataObject. Written exclusively in the Progress 4GL and SpeedScript compatible, the SmartDataObject allows you to define and manage business logic and data verification. See the *Progress AppBuilder Developer's Guide*for more information about SmartDataObjects.

2.1.5 Templates and Wizards

The AppBuilder provides templates and wizards to assist you when you create new Web objects, HTML files, and SmartDataObjects. From the main window of the AppBuilder, choose File→ New to see the list of templates and wizards that are available.

If you have a WebSpeed license, the Objects list displays only WebSpeed objects. If you have a license for the Unified AppBuilder product, the list also displays objects that you can use for Progress applications.

You can use the WebSpeed wizards to create Web objects without writing any HTML or SpeedScript code. The WebSpeed wizards prompt you for the necessary information and automatically generate the required code.

The WebSpeed wizards are:

- **Report**—Creates a tabular view of the database fields you specify. You can add navigation buttons for displaying the first, previous, next, and last set of records. You can also include a text entry field to allow the user to enter search criteria.
- **Detail**—Creates an updateable form to display database data, based on the selection criteria you specify. You can add transaction control buttons to allow users to add or delete records and to submit or reset changes.
- **HTML Mapping**—Maps the form fields in an existing HTML file to database fields.

In addition to the wizards, the following templates are available:

- **Main**—A Welcome or a static text page for a WebSpeed application.
- **CGI Wrapper**—A Web object that dynamically generates HTML within the context of SpeedScript.
- **Blank**—A basic HTML file that includes an embedded SpeedScript section.
- **Frameset**—An HTML file that creates three frames (a banner and two columns). The template also contains markup that displays a message when the browser does not support frames.
- **Report**—An HTML file with preprocessor definitions for creating a tabular view of database fields with navigation buttons.
- **Table**—A file for formatting database results into an HTML table. This is not a stand-alone file since it does not contain all the required HTML tags. It is meant to be inserted into another HTML file.

2.1.6 Editors

You can edit procedure files with conventional text editors, such as NotePad or the Progress Procedure Editor.

The Code Section Editor is the AppBuilder's main tool for editing structured procedure files. Unlike other editors. it is integrated with the AppBuilder and it preserves the structure of structured procedure files.

To invoke the Code Section Editor, choose the Code button or Window→ Code Section Editor from the AppBuilder main menu. You can also invoke the Code Section Editor by double-clicking on an object. (You have to enable double-clicking on objects in the Options→ Preferences window.)

See the *Progress AppBuilder Developer's Guide* for more information about the Code Section Editor. Also see the AppBuilder online help.

2.2 WebTools

WebTools are a collection of browser-based utilities that allow you to perform development tasks and to access information. Each tool is directly accessible by clicking the available link from the WebSpeed WebTools page.

The primary advantage of WebTools is that you do not have to install WebSpeed software locally in order to use them. You can access WebTools by specifying the WorkShop URL in your browser (*host_name*/scripts/cgiip.exe/WService=wsbroker1/workshop, for example).

However, if you have WebSpeed installed locally, you can also access WebTools from the Tools menu in AppBuilder. Select Tools→ WebTools from the AppBuilder main window.

As shown in Figure 2–2, the WebTools main page contains an index of links in the left frame and a work area in the right frame.

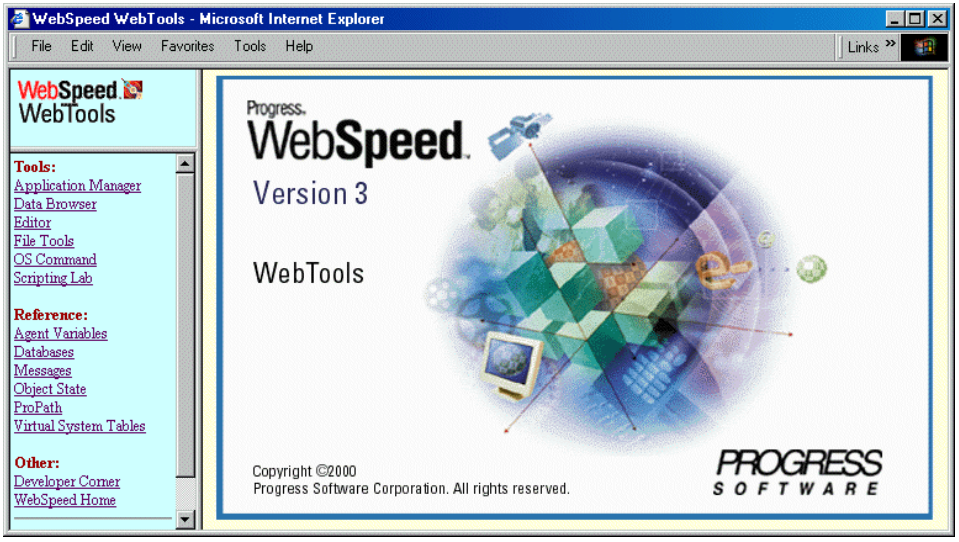


Figure 2–2: WebSpeed WebTools Main Page

The following sections give a brief description of each one of the WebTools. For more detailed information, see the online help, which is accessible from the main frame of each WebTool.

2.2.1 Application Manager

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.

2.2.2 Data Browser

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.

2.2.3 Editor

The Editor WebTool functions as a basic, single-document text editor. Only one file can be edited at a time. If you attempt to open a new file, the text-entry frame is checked for changes. If there are any changes, you are prompted to save or ignore those changes before a new file is opened.

2.2.4 File Tools

File Tools provides a directory listing from which you can view, open, run, compile, extract, or delete files in the selected directory. By default, File Tools initially displays a view of your Progress working directory. Once you have created some of your own Web objects and procedure files, you can use the File Tools options to manage and maintain files either in your Progress working directory or in another directory.

2.2.5 OS Command

The OS Command WebTool allows you to enter operating system commands. Commands execute in the operating system context of the machine where the WebSpeed Development Server is running. Command output is displayed below the command entry field.

NOTE: WebSpeed does not maintain context from one command to the next. Therefore, some commands (like changing your working directory, for example) do not execute. You get an error message stating that the command generated no output.

The OS Command WebTool allows you to run single-line command scripts. You can create these scripts in any text editor, but the file can only have one line (no carriage returns). Save them in *install_dir*`/src/web/scripts` with a .os file name extension. After you create .os files in that directory, their names appear in the OS Command WebTool frame. After you select a script and choose the Load Script button, the script's contents appear in the command entry field. Choose the Submit button to execute.

2.2.6 Scripting Lab

The Scripting Lab allows you to enter SpeedScript, SQL, JavaScript, VBScript, or HTML code and run it on the WebSpeed Development Server. You can use the Scripting Lab to test your code. You can copy your tested code into a Web object (by performing a cut-and-paste operation), but you cannot save code directly in the Scripting Lab.

See the *WebSpeed Developer's Guide* for information on programming with SpeedScript. See the *Pocket WebSpeed* guide for a list of all SpeedScript keywords.

2.2.7 Databases

The Databases WebTool allows you to look up information about the database schema of any database connected to the WebSpeed Agent. The Databases tool presents a view of the database on the server machine (which might be a different database from the one the AppBuilder is connected to). You can use the Databases tool to view:

- Databases connected to the WebSpeed Agent
- Table definitions for any database connected to the WebSpeed Agent
- Field and index definitions for a table
- Field attributes for a field
- Index attributes for an index

2.2.8 Agent Variables

The Agent Variables WebTool allows you to display the environment variables for your WebSpeed session. You can display any combination of the following categories of variables:

- CGI Environment variables
- WebSpeed variables
- WebSpeed WEB-CONTEXT attributes
- WebSpeed SESSION attributes

2.2.9 ProPath

The ProPath WebTool displays a list of the directories contained in the PROPATH environment variable. The PROPATH environment variable specifies where the WebSpeed Agent will search to locate the file it has been requested to run. You can use the ProPath tool to view and verify the directories included in the PROPATH. You can also use the ProPath WebTool to open a directory in the File Tools WebTool.

2.2.10 Virtual System Tables

The Virtual System Tables WebTool allows you to view the performance statistics that are saved in a database's virtual system tables.

For a general description of virtual system tables, see the *Progress Database Administration Guide and Reference*.

2.2.11 Object State

The Object State WebTool allows you to view the state of procedures that a WebSpeed Agent is running.

When you select an object state link, a table appears containing the name of each *persistent object* running on the WebSpeed Agent and a description of the type and state of the object.

2.2.12 Messages

The Messages WebTool allows you to look up the meaning of any WebSpeed error number.

Message descriptions include a link to the Progress KnowledgeBase. The KnowledgeBase is a service that provides access to WebSpeed's online technical support database. If you have a KnowledgeBase account, you can click this link to query the KnowledgeBase and selectively search for information on any topic, keyword, or error message and scan expert answers to the most frequently asked technical questions.

2.2.13 Home Pages

From the WebTools Detail frame you can access the Developer's Corner and WebSpeed home page.

The Developer's Corner is an information resource for Web programmers building ITP applications. The Developer's Corner provides useful links to:

- A resource center
- A support center
- Security information
- Frequently asked questions
- Installation information
- Interfaces

The WebSpeed home page provides:

- Product information
- Services and Support
- Customer profiles
- Press releases
- White papers
- Strategic partners
- WebSpeed demos

2.2.14 Online Help

From WebSpeed WebTools, you can access the Tools online help system:

- To get instructions for performing WebTools tasks, click the Help link. Then click the How to link. A list of each how-to topic, with a link to instructions for accomplishing a task, appears in a separate browser window.
- To look up a WebSpeed or Internet term, click the Help link. Then click the Glossary link. An indexed list of terms appears in a separate browser window.
- To view a description of each WebTool and access information about how to work with each tool, click the WebSpeed WebTools link.
- To access information about how to use the current tool or option, click the Help icon of any WebSpeed WebTool main frame.

To look up a SpeedScript language element, see the AppBuilder online help.

3

Setting Up the WebSpeed Development Environment

In this chapter you will learn how to set up your WebSpeed development environment to run the sample applications described in Chapter 4, “Sample WebSpeed Applications,” or to create the sample application described in Chapter 5, “Creating a WebSpeed Application.”

Contents of this chapter include:

- Creating and Connecting To a Database
- Setting AppBuilder Preferences For WebSpeed

3.1 Creating and Connecting To a Database

To run the sample program described in Chapter 4, "Sample WebSpeed Applications," or to complete the tutorial in Chapter 5, "Creating a WebSpeed Application," you must create a copy of the Sports2000 database in your working directory, start a server for it, and then connect to the server. The following sections guide you through the steps for completing these tasks.

3.1.1 Making a Working Copy Of the Sports2000 Database

You should always make a working copy the Sports2000 database rather than using the master copy that you installed with your program files. You can experiment freely with your working copy and, if necessary, restore it to its original state from the master copy.

Follow these steps to create a working copy of the Sports2000 database:

1 ♦ From the AppBuilder main menu, choose Tools→ Data Dictionary.

The Dictionary Startup dialog box, shown in Figure 3–1, appears.

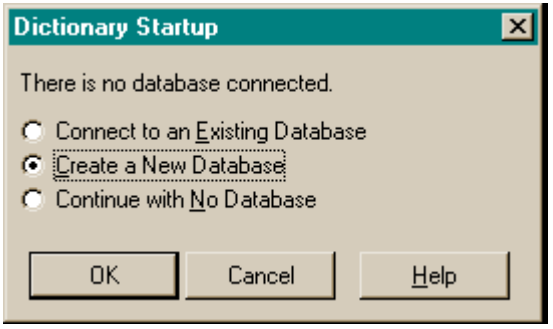


Figure 3–1: Dictionary Startup Dialog

2 ♦ Choose the Create a New Database radio button, then choose OK.

The Create Database dialog box appears.

3 ♦ In the New Physical Database Name field, type **Sports2000**.

4 ♦ Select the radio button labeled: A Copy of Some Other Database.

5 ♦ Choose Files and select `sports2000.db` in the folder where Progress is installed.

By default, `sports2000.db` is installed in `C:\Program Files\Progress\`. The completed Create Database dialog should look similar to Figure 3–2.

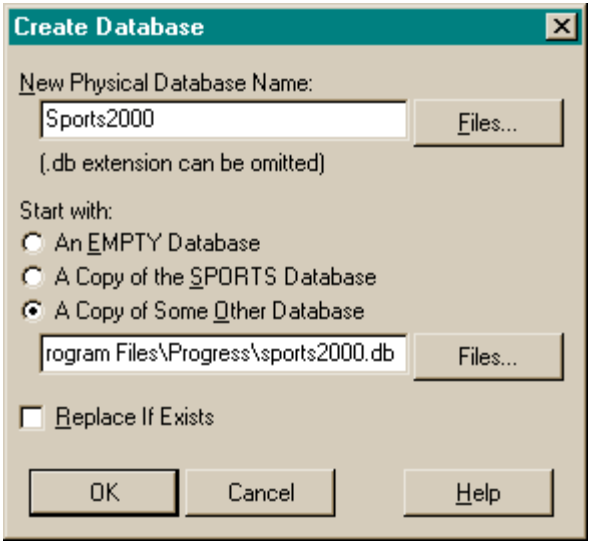


Figure 3–2: Create Database Dialog

6 ♦ Choose OK.

The Data Dictionary creates a copy of the Sports2000 database in your working directory (by default, `C:\Progress\Wrk`). The Connect Database dialog box appears, as shown in Figure 3–3. It contains the name of the database you just created.

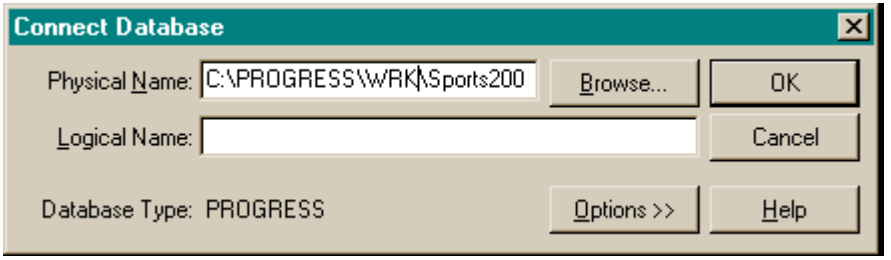


Figure 3–3: Connect Database Dialog

7 ♦ Choose Cancel.

You **do not** want to connect to the database at this time through the Data Dictionary. WebSpeed application development requires that both the AppBuilder and a WebSpeed Agent are connected to the database. Multiple connections are only possible if you start a server for the database. If you open a connection through the Data Dictionary now, the database is opened in single-user mode and it would be impossible to start a server for it.

8 ♦ From the Data Dictionary main menu, choose Database→ Exit.

The AppBuilder main window becomes the active window again.

3.1.2 Starting a Server For the Database

Both the AppBuilder and a WebSpeed Agent must be connected to the Sports2000 database before you can run sample applications or begin the tutorial in Chapter 5, "Creating a WebSpeed Application." Since the AppBuilder and the WebSpeed Agent function as individual users of the database, the database must be started in multi-user mode. To start a database in multi-user mode, you must start a server for the database.

Before you begin, you should make a copy of the Sports2000 database. The procedure is described in the "Making a Working Copy Of the Sports2000 Database" section.

The following steps show how to use Progress Explorer to start a server for the Sports2000 database. Progress Explorer is an administrative utility that runs on Windows platforms. For more information about Progress Explorer, see the *Progress Installation and Configuration Guide Version 9 for Windows*.

NOTE: You can also use the PROSERVE command to start a server for the database. For more information about PROSERVE, see the *Progress Startup Command and Parameter Reference*.

1 ♦ Make sure that an AdminServer is running on the system where you want to start a database server for your working copy of the Sports2000 database.

An AdminServer is installed on every system where you installed a Progress database, AppServer, NameServer, or WebSpeed Transaction Server.

Under the default configuration on Windows NT-based systems, the AdminServer starts automatically and runs as an NT service. You can start and stop the AdminServer from the Services control in the Windows Control Panel. For more information about running the AdminServer, see the *Progress Installation and Configuration Guide Version 9 for Windows*. For information on running an AdminServer on UNIX systems, see the *Progress Installation and Configuration Guide Version 9 for UNIX*.

NOTE: You might have to verify that a NameServer is also running. However, if you use the default configuration to start the AdminServer, the NameServer will be started automatically.

2 ♦ Choose Start→ Programs→ Progress→ Progress Explorer from your Windows Taskbar.

The Progress Explorer main window appears. When you expand the tree view under Progress Explorer, you see a list of machines where you can start a Progress server. In Figure 3–4, only the local host machine is listed.

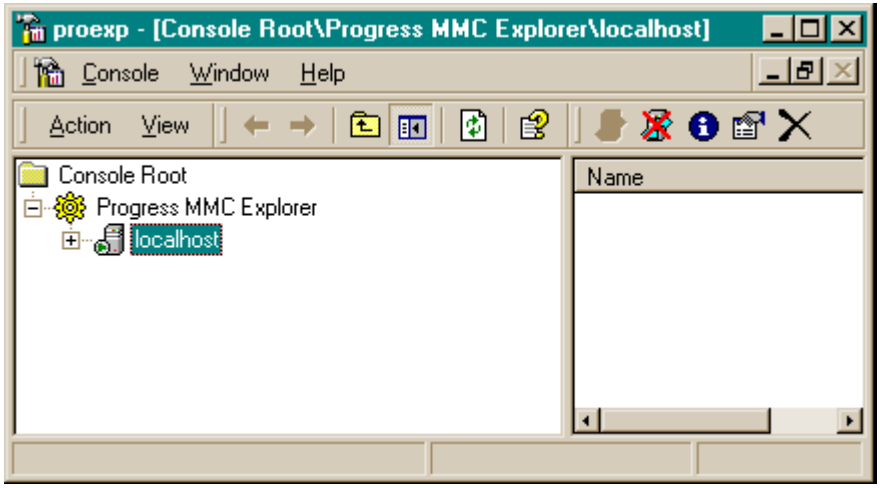


Figure 3–4: Progress Explorer Main View

If you need to add your local machine or a remote machine to the list, do the following:

a) Select Progress MMC Explorer from the tree view.

b) Choose Action→ Add Progress Server from the main menu.

c) Enter the server name (the name of a remote machine or **localhost**), your user name, and your password in the Server Properties dialog box.

d) Select Save username and password if you want to automatically log into this machine every time you connect to it.

e) Select Reconnect at startup only if you want to automatically connect to this machine every time you start Progress Explorer.

f) Choose OK.

3 ♦ Select the name of the machine where you created your working copy of the Sports2000 database.

4 ♦ Choose Action→ Connect from the main menu.

You will see a login dialog box, unless you configured Progress Explorer to automatically log into a particular machine.

5 ♦ Expand the tree view of the connected server machine.

If you are connected to the local host machine, the expanded tree view will look similar to Figure 3–5.

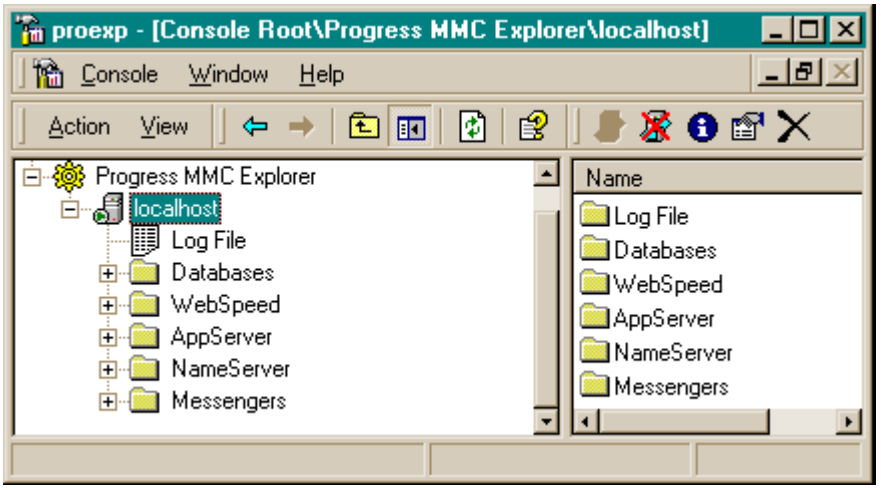


Figure 3–5: Progress Explorer Expanded Tree View

6 ♦ Select the node labeled Databases in the tree view.

7 ♦ Choose Action→ New.

As shown in Figure 3–6, the New Object dialog box appears.

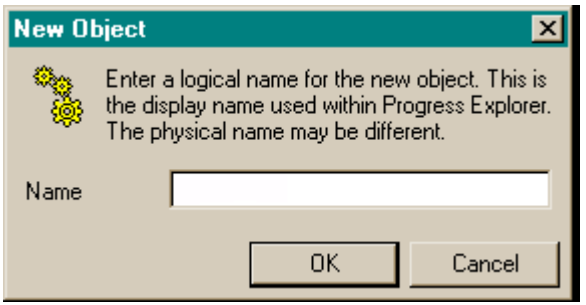


Figure 3–6: New Object Dialog In Progress Explorer

8 ♦ Enter **Sports2000** and choose OK.

The Database Properties dialog box appears and looks similar to Figure 3–7.

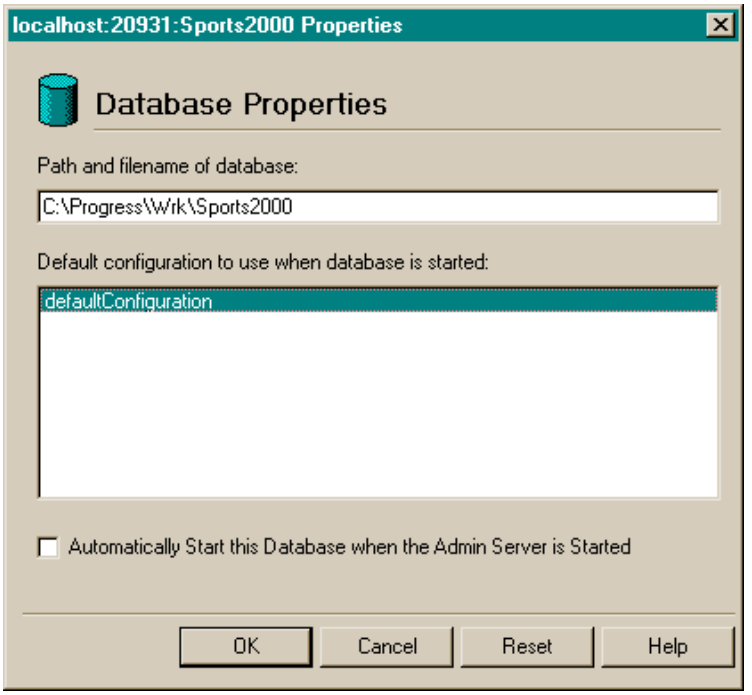


Figure 3–7: Database Properties Dialog In Progress Explorer

9 ♦ Enter the pathname of the Sports2000 database in the Path and filename field and choose OK.

As shown in Figure 3–8, icons labeled Sports2000 appear in the Progress Explorer window.

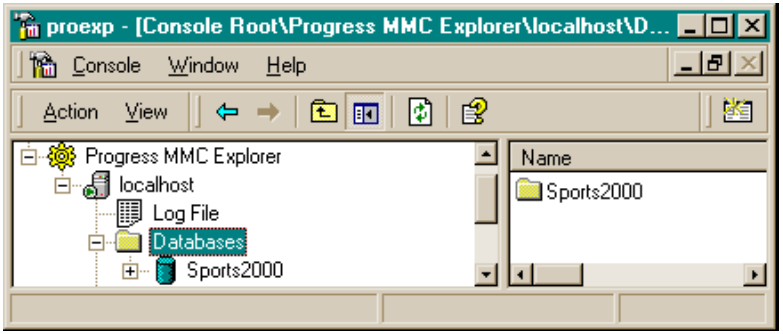


Figure 3–8: Sports2000 In Progress Explorer

10 ♦ Specify a service name or port number for the database.

To specify a service name or port number in Progress Explorer:

a) Expand the tree view under Configurations until you see the server group node. (For example, under defaultConfiguration the server group node might be labeled as defaultServerGroup.)

b) Select the server group for the configuration that you want to start.

c) Choose Action→ Properties.

d) Enter a service name or port number in the properties dialog box.

e) Choose OK.

For more information about specifying service names and port numbers, consult with your system administrator. Also see the *Progress Installation and Configuration Guide Version 9 for Windows* or the *Progress Installation and Configuration Guide Version 9 for UNIX*.

11 ♦ Select the Sports2000 icon and choose Action→ Start Sports2000.

Progress Explorer starts a server for the Sports2000 database. You can verify that the server is running by choosing Action→ Status of Sports2000.

Do not exit from Progress Explorer at this time. You can continue your session in Progress Explorer in order to start a WebSpeed Broker, which is described in the section, "Starting a WebSpeed Broker With Agents Connected To the Database."

3.1.3 Starting a WebSpeed Broker With Agents Connected To the Database

This section describes how to use the Progress Explorer to configure and start a WebSpeed Broker with Agents connected to your working copy of the Sports2000 database.

Before you begin, you should:

- Make a copy of the Sports2000 database. The procedure is described in the "Making a Working Copy Of the Sports2000 Database" section.
- Start a database server for your copy of the Sports2000 database. The procedure is described in the "Starting a Server For the Database" section.

NOTE: The following procedure assumes that you have started Progress Explorer and have connected to the appropriate host machine. If you have not started and connected Progress Explorer, follow the first five steps in the "Starting a Server For the Database" section.

To start a WebSpeed Broker and WebSpeed Agents, complete the following steps in Progress Explorer:

1 ♦ Select and expand the node labeled WebSpeed in the tree view.

As shown in Figure 3–9, you should see the default broker name, wsbroker1, in the tree view.

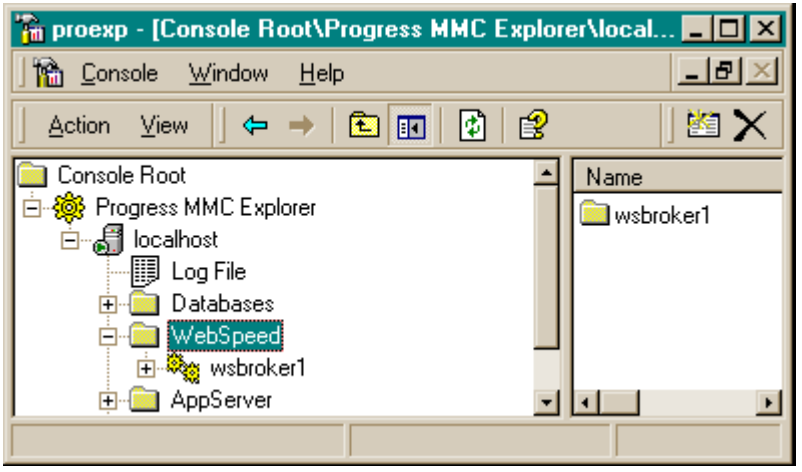


Figure 3–9: wsbroker1 In Progress Explorer

2 ♦ Select wsbroker1 (from either the right or the left pane of the window) and choose Action→ Properties.

The Properties dialog box appears with a tree view in the left pane.

3 ♦ Expand the tree view under Agent and select General.

The Properties dialog box looks similar to Figure 3–10.

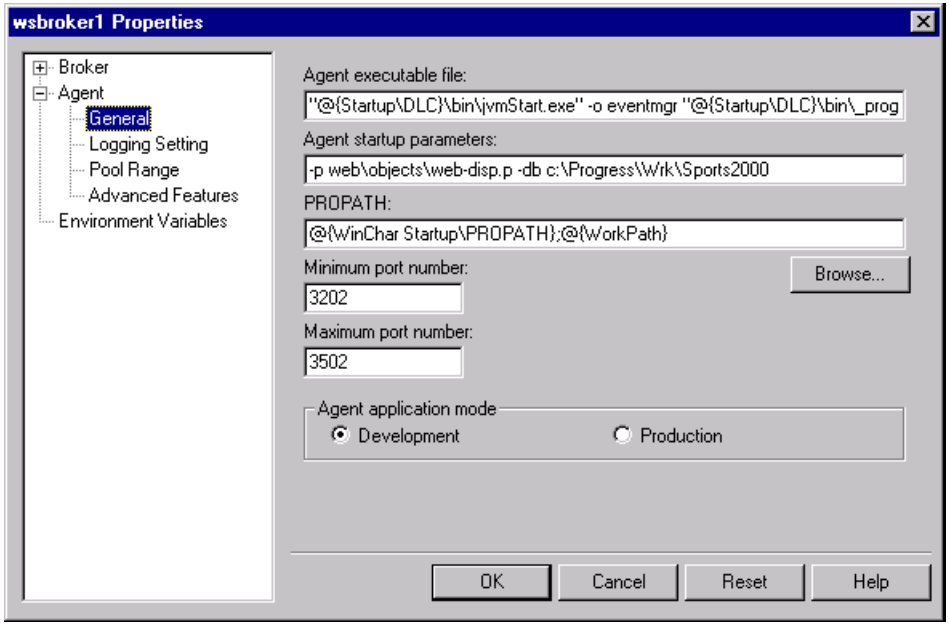


Figure 3–10: Broker and Agent Properties In Progress Explorer

4 ♦ Add the pathname of your working copy of the Sports2000 database to the agent startup parameters using the -db option.

In the figure in Step 3, the database is specified as `c:\Progress\Wrk\Sports2000`.

NOTE: In this example the database is located on the same machine as the WebSpeed Broker. If the server for the database is running on a remote machine, you must specify the host name (using the -H parameter) of the remote machine and the service name (using the -S parameter). For more information about connecting to remote databases, see the *WebSpeed Installation and Configuration Guide*.

5 ♦ Choose OK.

6 ♦ Select wsbroker1 and choose Action→ Start.

After the WebSpeed Broker starts, you can monitor its performance by choosing Action→ Status.

3.1.4 Connecting the AppBuilder To the Database

In addition to connecting a WebSpeed Agent to your working copy of the Sports2000 database, you also need to connect AppBuilder to the same database.

Before you begin, you should:

- Make a copy of the Sports2000 database. The procedure is described in the "Making a Working Copy Of the Sports2000 Database" section.
- Start a database server for your copy of the Sports2000 database. The procedure is described in the "Starting a Server For the Database" section.

Follow these steps to connect the AppBuilder to your working copy of the Sports2000 database after you have started a server for it:

1 ♦ From the AppBuilder main window, choose Tools→ Database Connections.

As shown in Figure 3–11, the Database Connections dialog appears.

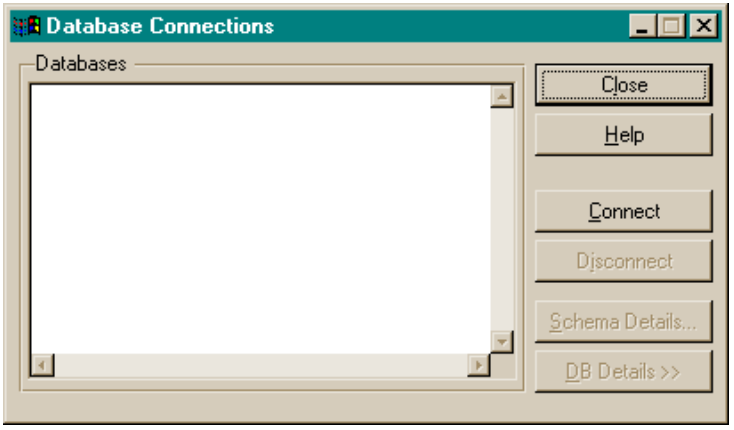


Figure 3–11: Database Connections Dialog In AppBuilder

2 ♦ Choose Connect.

The Connect Database dialog box appears.

3 ♦ Choose Options.

The Connect Database dialog box expands to display database connection options.

4 ♦ Select the Multiple Users check box.

5 ♦ Enter the pathname of the Sports2000 database in the Physical Name field.

After you complete this step, the Connect Database dialog box should look similar Figure 3–12.

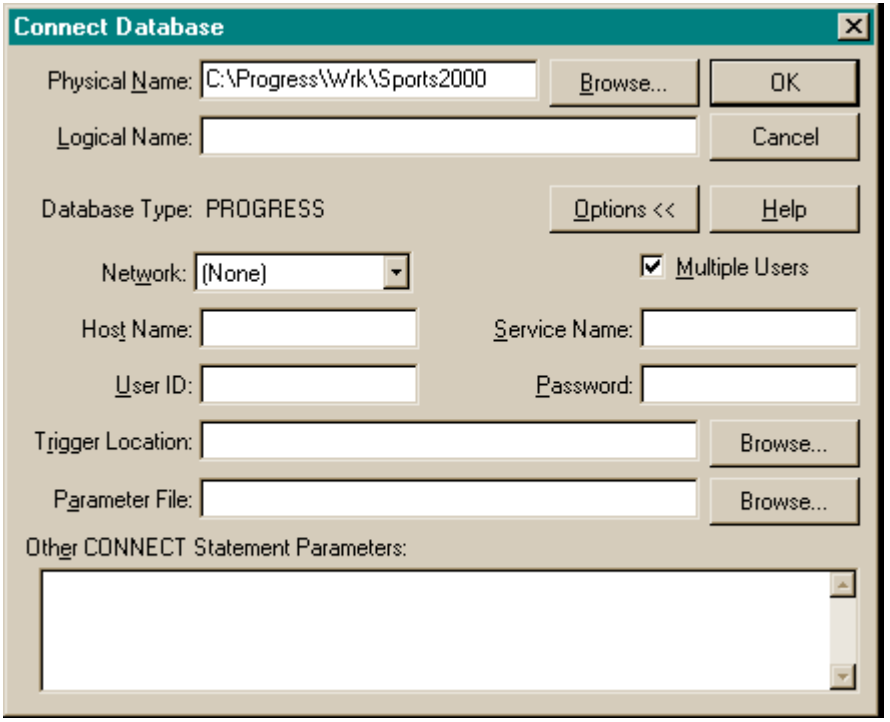


Figure 3–12: The Connect Database Dialog

6 ♦ Choose OK.

The Database Connections property sheet appears with Sports2000 in the list of database connections.

3.2 Setting AppBuilder Preferences For WebSpeed

If you want to run and test the Web objects that you create with WebSpeed, you must specify which WebSpeed Broker you want the AppBuilder to use. You must also specify which browser you want the AppBuilder to use for viewing the Web pages that Web objects generate.

Before you begin to set these AppBuilder preferences, check the following:

- Check that you have installed and configured the WebSpeed Server.

  See the *WebSpeed Installation and Configuration Guide* for more information.

- Make sure that a database server is running for your working copy of the Sports2000 database.

  See the "Making a Working Copy of the Sports2000 Database" and the "Starting a Server For the Database" sections.

- Make sure that your WebSpeed Broker is running and is connected to your working copy of the Sports2000 database.

  See the "Starting a WebSpeed Broker With Agents Connected To the Database" section.

- Make sure that the AppBuilder is running and is connected to your working copy of the Sports2000 database.

  See the "Connecting the AppBuilder To the Database" section.

Follow these steps to set AppBuilder preferences for WebSpeed application development:

1 ♦ From the AppBuilder main window, choose Options→ Preferences.

The Preferences dialog box appears.

2 ♦ Select the General tab to view the available options.

You should not have to modify the settings on the General tab in order to complete the tutorial in Chapter 5, "Creating a WebSpeed Application." It is important, however, that you understand how the Qualify Database Fields with Database Name setting works, as it might be enabled by default.

If this setting is enabled, WebSpeed prepends references to database fields with the database name when generating SpeedScript code. For example, the Customer.Name field would appear in all WebSpeed-generated SpeedScript code as sports2000.Customer.Name.

You can disable this setting to prevent WebSpeed from prepending references to database fields with the database name. Unless you use multiple databases, you can disable this setting.

Choose the Save Settings button to save any changes you make.

3 ♦ Select the WebSpeed tab.

The WebSpeed Preferences dialog, shown in Figure 3–13, appears.

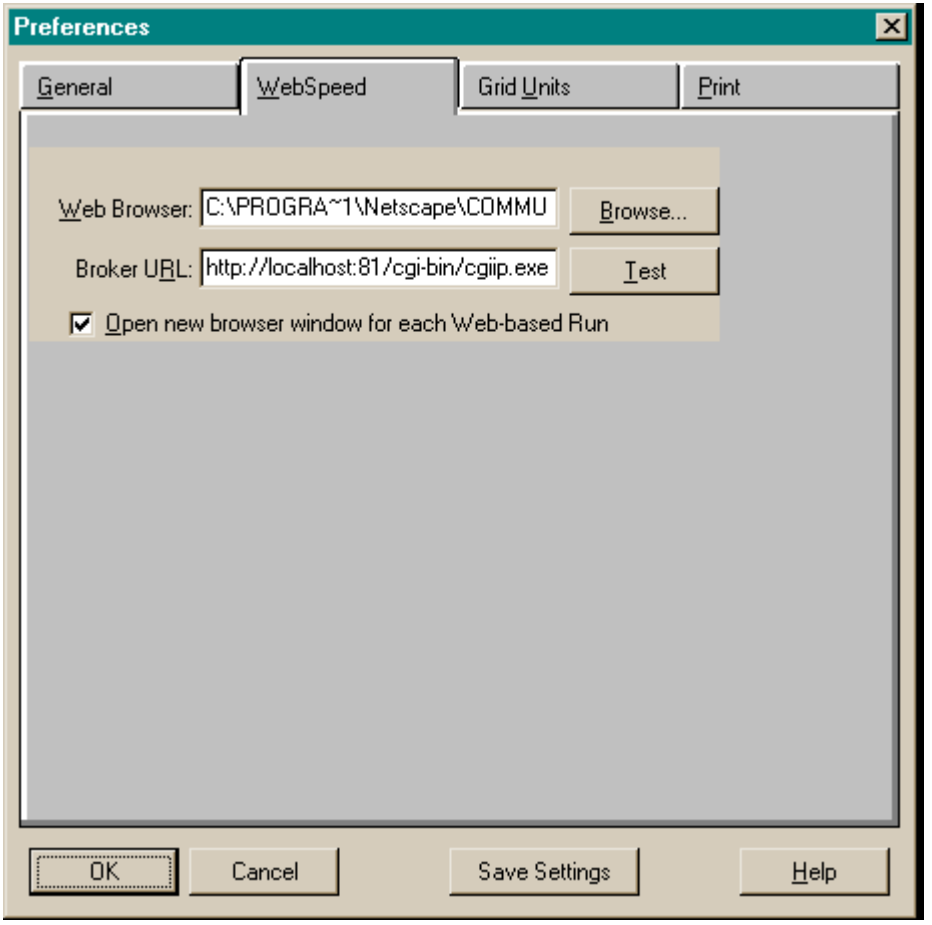


Figure 3–13: WebSpeed Preferences Dialog

4 ♦ Specify the pathname of the browser that you intend to use during development.

For example, `C:\Program Files\Netscape\Communicator\Program\netscape.exe` and `C:\Program Files\Plus!\Microsoft Internet\iexplore.exe` are typical pathnames for Netscape Communicator and Internet Explorer respectively.

5 ♦ Specify the URL for a WebSpeed Broker.

The URL actually specifies the location of the WebSpeed Messenger that connects to the WebSpeed Broker. The WebSpeed Messenger resides in the scripts directory of your WebServer.

The syntax for specifying the URL is:

```
http://host_name[:port]/scripts_dir/messenger/WService=broker_name
```

For more information about WebSpeed URLs, see the *WebSpeed Developer's Guide*.

6 ♦ Activate the Open new browser window toggle box if you want each Web object to run in a new browser window.

This feature is useful if you want to compare output after making changes to your source code. Otherwise, WebSpeed reuses a single browser window.

If you are using the Netscape browser, you might want to start new windows when testing Web objects. The Netscape browser cache is not always refreshed, and the results shown in the browser can be stale if you are reusing the same browser window.

7 ♦ Choose the Test button to test the connection to the WebSpeed Broker.

A browser opens displaying either a success or a failure to connect message. With a successful connection, you will also see information about the broker, such as its working directory, the PROPATH settings, and the names of any connected databases.

8 ♦ Choose the Save Settings button to save the Web Setup preferences for future AppBuilder sessions.

9 ♦ Choose OK to close the Preferences dialog box and return to the AppBuilder main window.

3.2.1 Setting the Development Mode

The WebSpeed AppBuilder supports two development modes: local and remote.

Use local mode when all of the following conditions are true:

- The WebSpeed AppBuilder is running on the same machine as the WebSpeed Development Server.
- Both the WebSpeed AppBuilder and the WebSpeed Development Server have the same working directory.
- Both the WebSpeed AppBuilder and the WebSpeed Development Server have the same PROPATH settings.

The working directory and PROPATH are set during installation of WebSpeed. For more information, see the *WebSpeed Installation and Configuration Guide*.

Use remote mode when the WebSpeed Development Server is on a remote machine, when the WebSpeed Development Server and the AppBuilder have different PROPATH settings, or when the WebSpeed Development Server has a different working directory than the AppBuilder. In remote mode, files are saved in WebSpeed Development Server's working directory on the remote machine.

To change the development mode, click on the Development Mode toggle button, which is the last button in the AppBuilder tool bar. The Development Mode button in remote and local modes is shown in Figure 3–14.

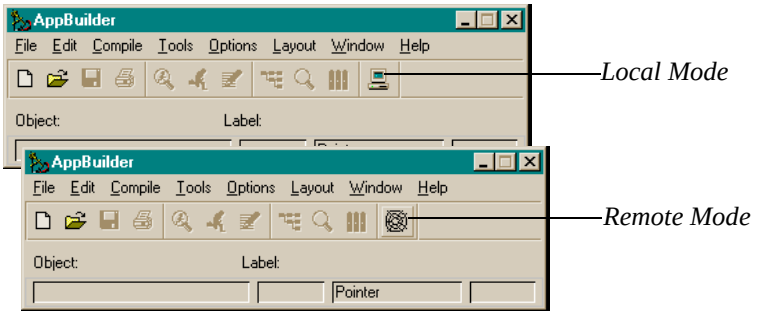


Figure 3–14: The Development Mode Button

4

Sample WebSpeed Applications

This chapter describes how to run the sample applications that are included with WebSpeed. In addition, it shows some of the features of SportsPro Sales Advisor, which is one of the sample applications.

Chapter 5, “Creating a WebSpeed Application” contains a tutorial that shows you how to use WebSpeed to create the SportsPro Sales Advisor.

Contents of this chapter include:

- Running Sample Applications
- Running SportsPro Sales Advisor
- Source Code For Sample Applications

4.1 Running Sample Applications

WebSpeed includes sample applications that illustrate intranet, internet, and extranet applications.

The sample applications run against the sample Sports2000 database, which is located in your installation directory. In order to run the sample applications, you should complete the following setup procedure:

1 ♦ Copy the Sports2000 database from the installation directory into your personal working directory.

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

3 ♦ Start a WebSpeed Broker connected to your working copy of the Sports2000 database.

4 ♦ Connect the AppBuilder to your working copy of the Sports2000 database.

5 ♦ Specify the WebSpeed Broker URL and your default browser in the AppBuilder Preferences dialog box.

You can find detailed instructions for completing this setup procedure in Chapter 3, "Setting Up the WebSpeed Development Environment."

To access the sample applications from the AppBuilder, choose Tools→ WebTools from the AppBuilder menu bar. Then select Sample Applications from the top of the main help page. Then select one of the following to run:

- **Sales Advisor Intranet Application**—An intranet application that you can create by following the tutorial in Chapter 5, "Creating a WebSpeed Application."
- **SportsPro Shopping Cart Internet Application**—An example of one of the most popular types of Web-based Internet applications.
- **Supplier Extranet Application**—An example of a business-to-business extranet application with some advanced WebSpeed features.

4.2 Running SportsPro Sales Advisor

This section illustrates some of the features of the SportsPro Sales Advisor application. Chapter 5, "Creating a WebSpeed Application," contains a tutorial that shows you how to use WebSpeed to build SportsPro Sales Advisor.

SportsPro Sales Advisor is an intranet application that supplies helpful information to sales reps of the fictional SportsPro sporting goods company. The SportsPro Sales Advisor allows sales reps to check inventory and maintain customer account information in the Sports2000 database, directly from their Web browsers.

Follow these steps to run SportsPro Sales Advisor:

1 ♦ Complete the setup procedures described in the "Running Sample Applications" section.

2 ♦ From the AppBuilder main window, select Tools→ WebTools. The WebSpeed WebTools page opens in a browser window.

3 ♦ In the WebTools index of links, click the Help link.

4 ♦ Click the Sample Applications link.

As shown in Figure 4–1, the Sample Applications home page opens in a separate browser window.

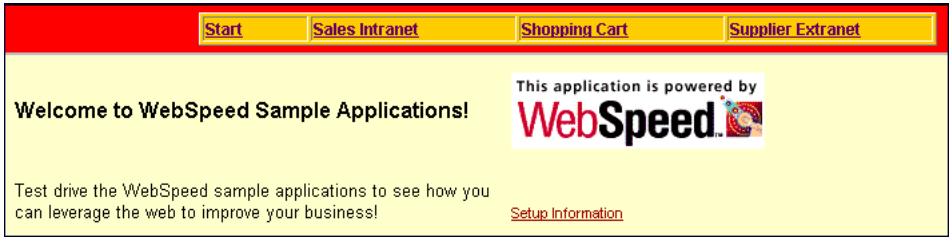


Figure 4–1: Sample Applications Home Page

5 ♦ Click the Sales Intranet link.

The SportsPro Sales Advisor application launches in your Web browser and displays an initial frame set that contains three separate Web objects: an application header (SportsPro Sales Advisor), an application index (the Inventory Info. and the Customer Info. links), and a welcome screen (Welcome to SportsPro Sales Advisor). The initial frame set is shown in Figure 4–2.



Figure 4–2: Sports Pro Sales Advisor Initial Frame Set

To view the HTML source file of any Web object, select the object and choose the View Code button.

6 ♦ Click the Inventory Info. link.

The Inventory Report , shown in Figure 4–3, opens in the Main frame and displays data from the Item table of the Sports2000 database.

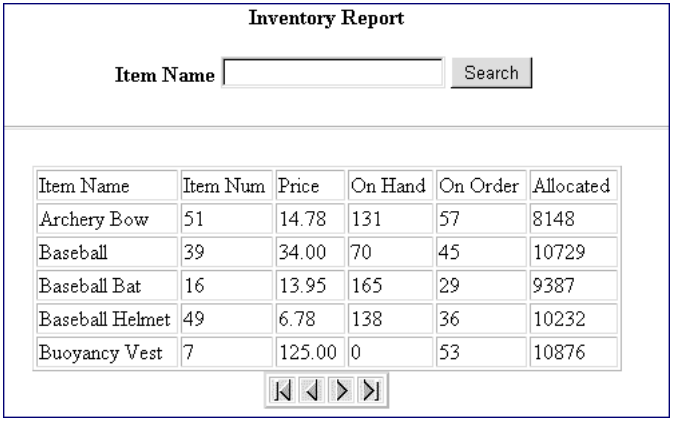
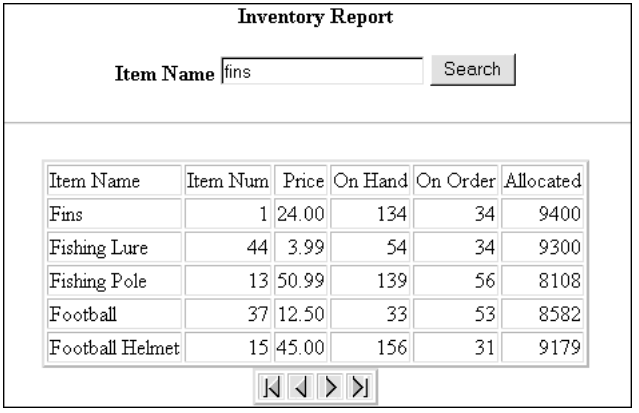


Figure 4–3: Inventory Report

Because the report is sorted by the Item Name field, the Item records appear in alphabetical order under Item Name. The navigation panel (the four directional buttons) allows you to scroll backward and forward through the Item records. You can also search for an Item Name value.

7 ♦ Type **fins** in the Item Name text entry field, then choose Search.

The Item record for Fins appears in the first row of the report:



8 ♦ Click the Customer Info. link.

The Customer Report, shown in Figure 4–4, opens in the Main frame and displays data from the Customer table of the Sports2000 database.

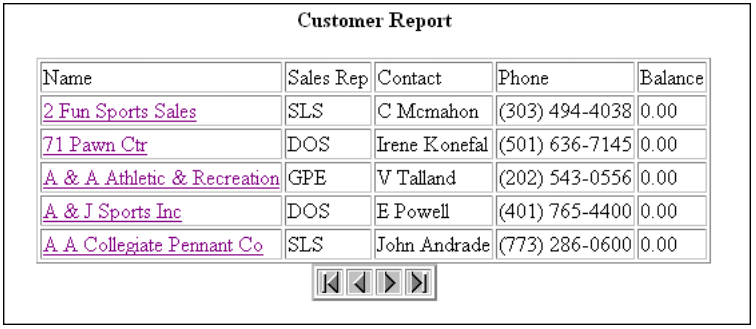


Figure 4–4: Customer Report

The navigation panel allows you to scroll backward and forward through the Customer records. The Name field provides a link to a Customer Update form.

9 ♦ Click a Name link.

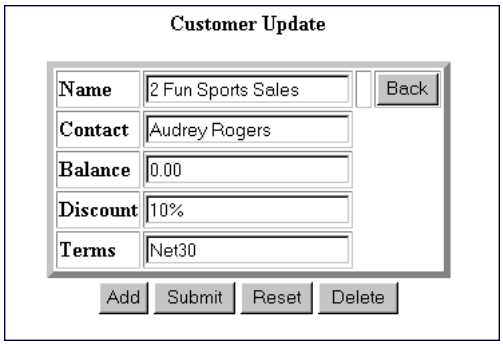
The Customer Update form, shown in Figure 4–5, opens in the Main frame and displays details for the selected customer.



Figure 4–5: Customer Update Form

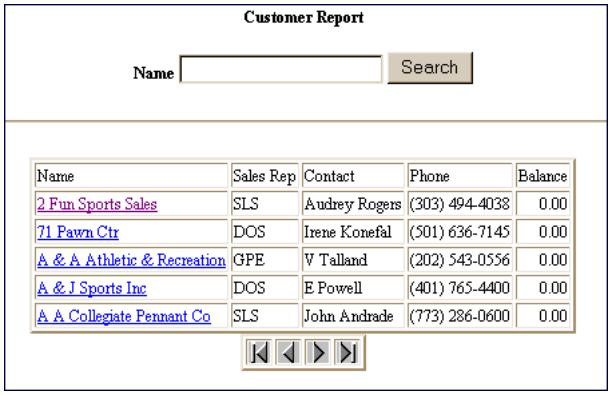
10 ♦ Change the Contact name, then choose Submit.

The Customer record is updated:



11 ♦ Choose Back to return to the Customer Report.

The new Contact name appears in the report:



4.3 Source Code For Sample Applications

You can find source files used to create the Web objects and procedure files for the sample applications in the *install-path*/src/samples/web directory.

5

Creating a WebSpeed Application

This chapter is a tutorial that shows you how to create a WebSpeed application by using WebSpeed wizards and templates to create Web objects. The Web objects you will build are the components of the SportsPro Sales Advisor intranet application, which is described in Chapter 4, "Sample WebSpeed Applications."

This chapter contains the following topics:

- Setting Up the WebSpeed Development Environment
- Creating a Report With Search Capability
- Creating a Master Report Linked To a Detail Update Form
- Using the Detail Wizard
- Working With Templates
- Where To Go From Here

5.1 Setting Up the WebSpeed Development Environment

Before you create the Web application described in this chapter, you must complete the following setup procedure:

1 ♦ Copy the Sports2000 database from the installation directory into your personal working directory.

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

3 ♦ Start a WebSpeed Broker connected to your working copy of the Sports2000 database.

4 ♦ Connect the AppBuilder to your working copy of the Sports2000 database.

5 ♦ Specify the WebSpeed Broker URL and your default browser in the AppBuilder preferences dialog box.

You can find detailed instructions for completing this setup procedure in Chapter 3, "Setting Up the WebSpeed Development Environment."

5.2 Creating a Report With Search Capability

The following steps guide you through creating a sample inventory report. The inventory report displays information about each item in inventory and allows sales representatives to search for information by item name.

As you go through the report creation, you will notice that you skip over certain options or click past one or more pages in the HTML Report Wizard without making any selections. For the purposes of creating this inventory report, it is not necessary for you to do anything other than what is described in the steps provided.

5.2.1 Starting the Report Wizard

1 ♦ From the AppBuilder main window, choose File→ New.

The New dialog, shown in Figure 5–1, appears.

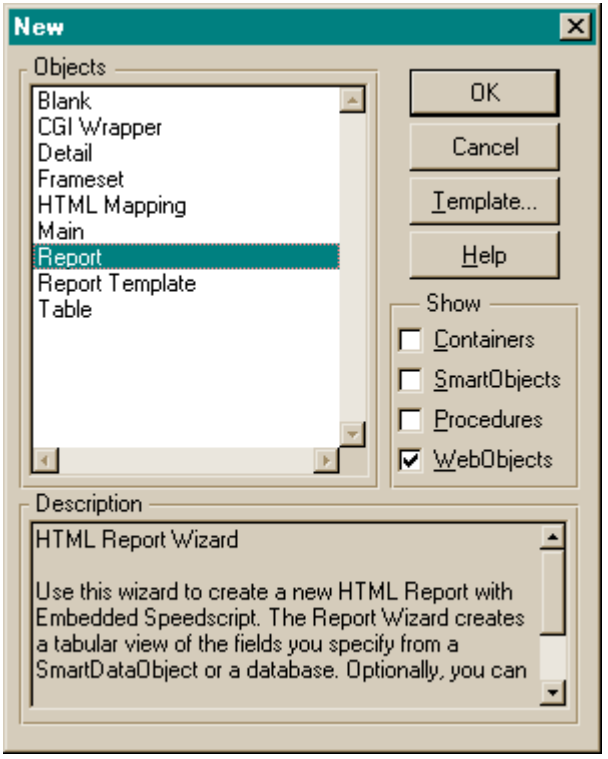


Figure 5–1: The New Dialog

2 ♦ Choose Report, and choose OK. Page 1 of the HTML Report Wizard opens.

3 ♦ Choose the Next button. Page 2 of the HTML Report Wizard opens.

5.2.2 Defining the Query

Page 2 of the Report Wizard allows you to choose whether to base your report on an existing SmartDataObject or on a database query. Follow these steps to define a database query for the inventory report:

1 ♦ Choose the Database option, then choose the Define Query button.

The Query Builder dialog, shown in Figure 5–2, appears.

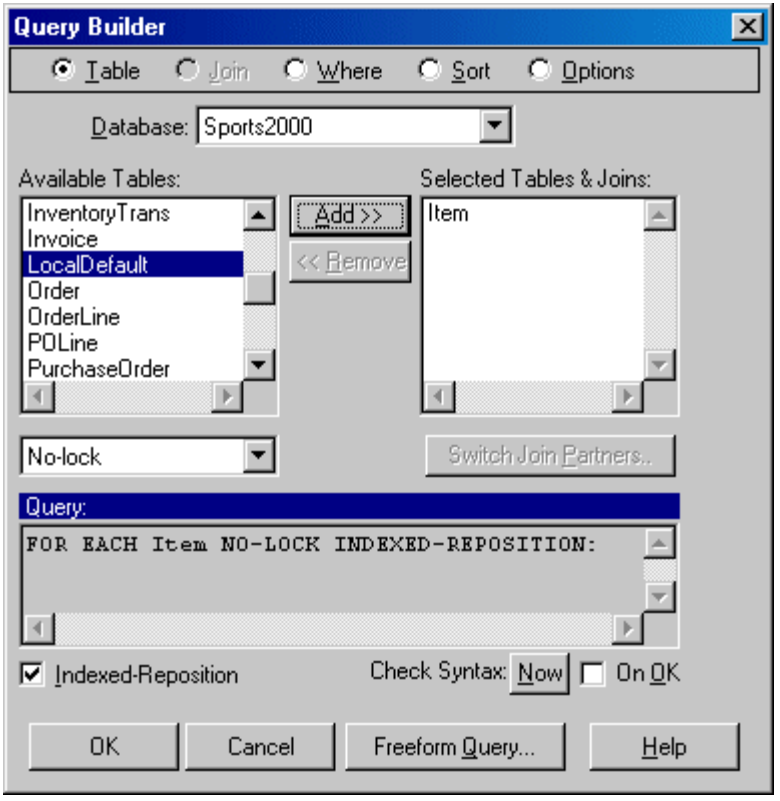


Figure 5–2: The Query Builder Dialog

2 ♦ Select Item from the Available Tables selection list, then choose the Add button to move it into the Selected Tables & Joins selection list.

3 ♦ Choose OK to close the Query Builder and return to Page 2 of the HTML Report Wizard.

4 ♦ Choose the Next button. Page 3, the External Tables page, appears.

5.2.3 Selecting Fields For the Query

1 ♦ Choose the Next button on the External Tables page. (You will not be selecting any External Tables to complete this report.)

Page 4, shown in Figure 5–3, appears and allows you to select the fields for your query:

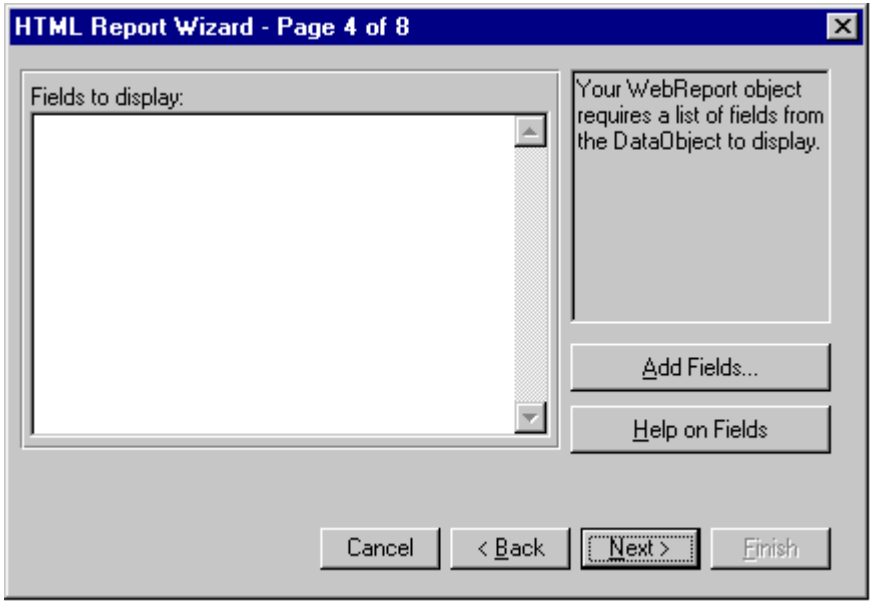


Figure 5–3: Page 4 (Fields) Of the HTML Report Wizard

2 ♦ Choose the Add Fields button. The Column Editor dialog box appears.

3 ♦ Choose the Add button. The Multi-Field Selector dialog box appears.

4 ♦ Select the following fields from the Available Fields selection list: Allocated, Onhand, OnOrder, ItemName, Itemnum, and Price.

5 ♦ Choose the Add button to move the fields into the Selected Fields list. In the report, fields appear in the same order as in the Selected Fields list.

6 ♦ To change the order of the Selected Fields list, select a field, then use the Move Up and Move Down buttons to move the fields in the list.

The fields should appear in the report in the following order: ItemName, Itemnum, Price, Onhand, OnOrder, and Allocated, as shown in Figure 5–4.

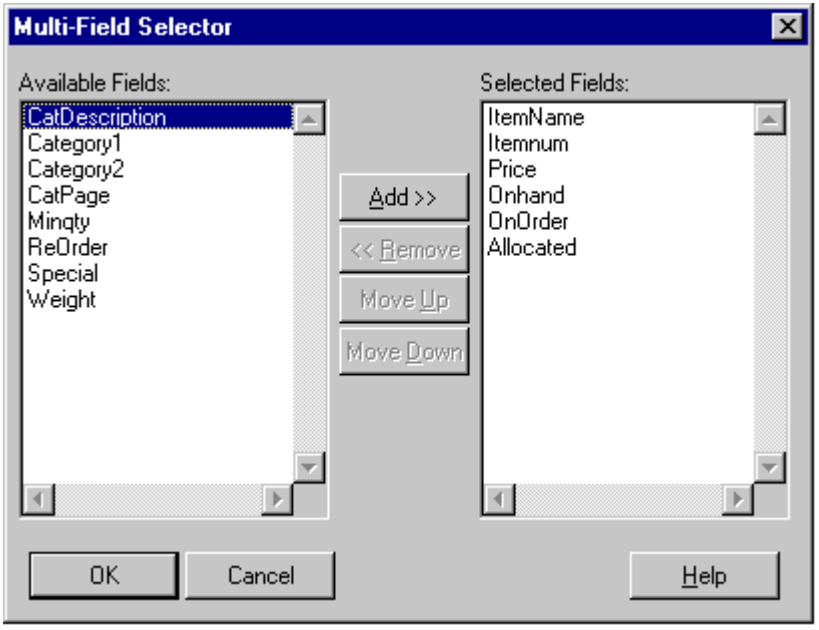


Figure 5–4: The Multi-Field Selector Dialog

7 ♦ Choose OK, then choose OK on the Column Editor dialog box.

Page 4 reappears, with the selected fields listed.

8 ♦ Choose the Next button.

Page 5, the Style page, appears.

5.2.4 Defining the Style Of the Report

Follow these steps from the Style page to customize the layout of your report:

1 ♦ In the Title field, type **Inventory Report**.

2 ♦ Type 5 in the Rows box.

3 ♦ Select 1 in the Border combo box.

4 ♦ If it is not already selected, select Search Form; then select the ItemName field from the Field selection list.

The completed page should look similar to Figure 5–5.

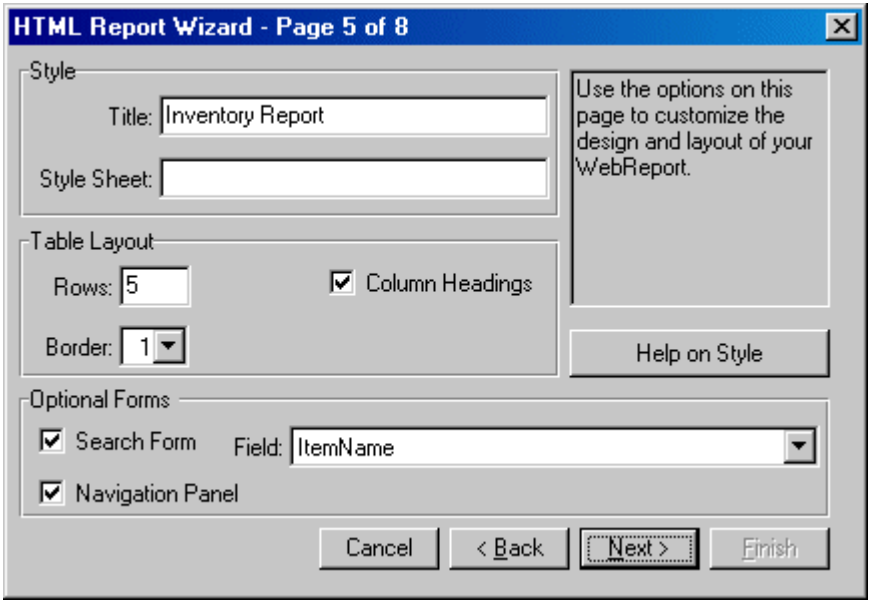


Figure 5–5: Page 5 (Style) Of the HTML Report Wizard

5 ♦ Choose the Next button. Page 6, the HyperLink page, appears.

6 ♦ Choose the Next button on the HyperLink page. (You will not be setting any links in this tutorial.)

Page 7, the Background page, appears.

7 ♦ Follow these steps to choose background colors:

a) In the Page Background color selection table, select a color.

b) In the Table Background color selection table, select a color.

Figure 5–6 shows the Background page with grey selected for the page color and white selected for the table color.

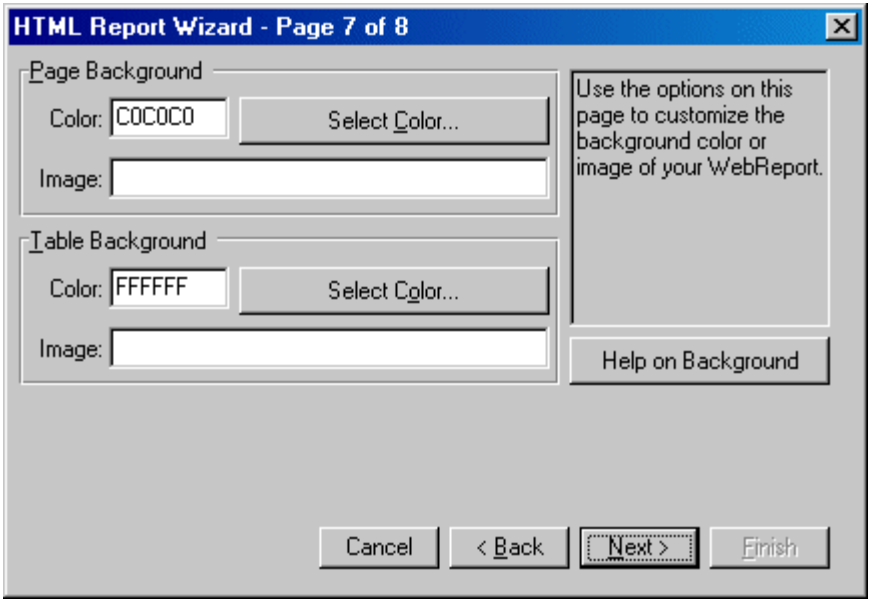


Figure 5–6: Page 7 (Background) Of the HTML Report Wizard

8 ♦ Choose the Next button.

Page 8, the last page in the wizard, appears.

5.2.5 Previewing and Saving the Report

Follow these steps to save the report you have created with the Report wizard:

1 ♦ On Page 8 of the Wizard, type **myinvrpt.html** in the Filename field.

2 ♦ Choose the Preview button.

The report is saved remotely and is converted from HTML to a Web object, which is then compiled remotely. As shown in Figure 5–7, the report is run and the results appear in a browser window.

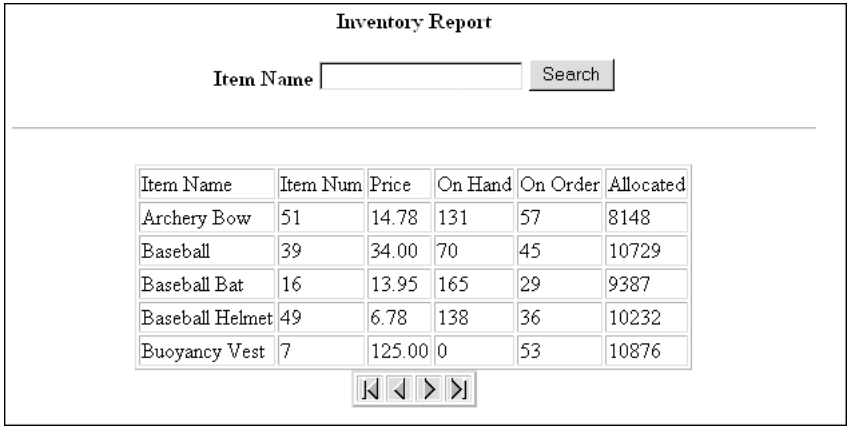


Figure 5–7: Inventory Report

3 ♦ Close the browser window.

4 ♦ Choose the Finish button in the wizard to complete that report. A Procedure Window appears, showing you the HTML source code for the object you have successfully created.

5 ♦ Close the Procedure Window.

5.3 Creating a Master Report Linked To a Detail Update Form

The customer report displays information about each customer and allows the sales representatives to link to a detail form, where they can update information for any customer. You use the Report Wizard to create the customer report. After you have created the customer report, you will create the customer detail update form.

Follow these steps to launch the Report Wizard:

1 ♦ From the AppBuilder main window, choose File→ New. The New dialog box appears.

2 ♦ Choose Report and choose OK. Page 1 of the HTML Report Wizard opens.

3 ♦ Choose the Next button. Page 2 appears.

5.3.1 Defining the Query

Follow these steps to define a query for the customer report:

1 ♦ Choose the Database option; then choose the Define Query button. The Query Builder appears.

2 ♦ Select Customer from the Available Tables selection list, then choose the Add button to move it into the Selected Tables & Joins selection list.

3 ♦ Choose OK.

Page 2, shown in Figure 5–8, reappears.

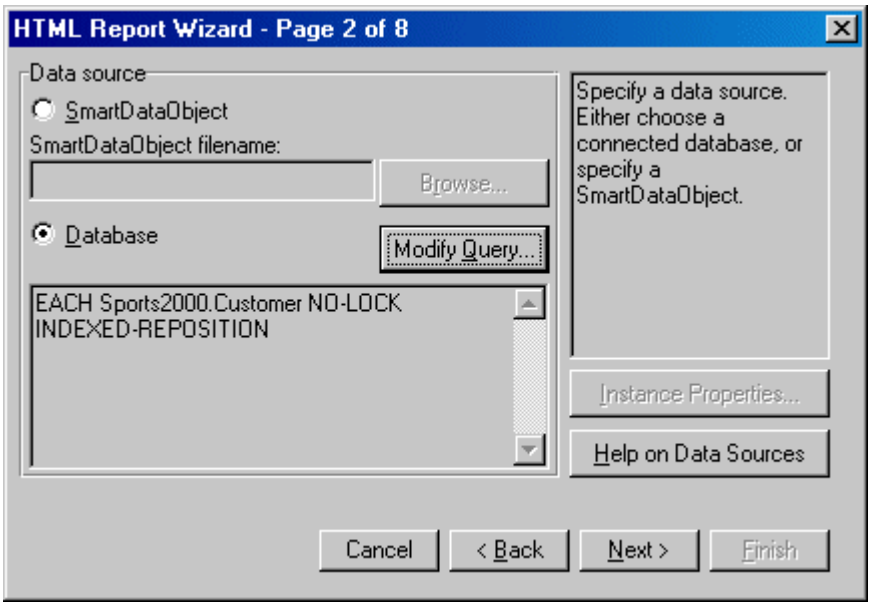


Figure 5–8: Page 2 (Data Source) Of the HTML Report Wizard

4 ♦ Choose the Next button. Page 3 (External Tables) appears.

5.3.2 Selecting Fields For the Query

Follow these steps to select the fields for the query:

1 ♦ Choose the Next button. (You will not be selecting any External Tables to complete this report.)

Page 4 (Fields) appears.

2 ♦ Choose the Add Fields button. The Column Editor appears.

3 ♦ Choose the Add button. The Multi-Field Selector dialog box appears.

4 ♦ Select the following field names from the Available Fields selection list: Name, SalesRep, Contact, Phone, and Balance.

5 ♦ Choose the Add button to move each field name to the Selected Fields list.

6 ♦ Choose OK. The Column Editor dialog reappears.

The Column Editor should look similar to Figure 5–9.

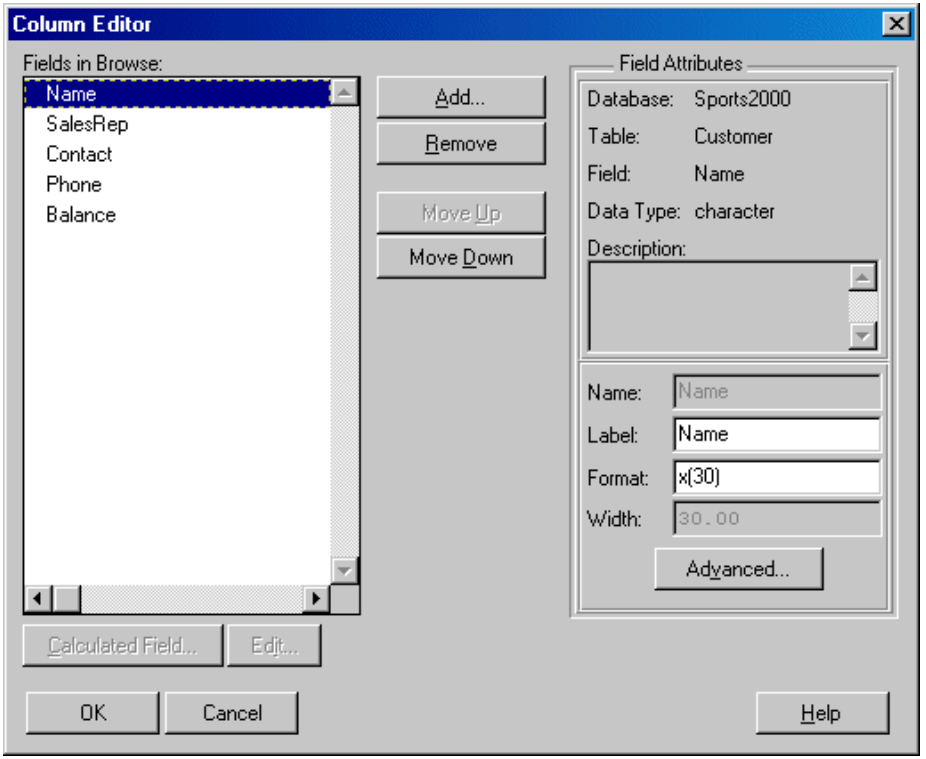


Figure 5–9: Column Editor

5.3.3 Defining the Style Of the Report

Follow these steps to define the report's style:

1 ♦ Choose OK. Page 4 reappears.

2 ♦ Choose the Next button. Page 5 appears.

3 ♦ In the Title field, type **Customer Report**.

4 ♦ Type **5** in the Rows box.

5 ♦ Select 1 in the Border combo box.

6 ♦ If it is not already selected, select Search Form.

7 ♦ In the Field combo box, choose Name.

After you complete these steps, page 5 should look similar to Figure 5–10.

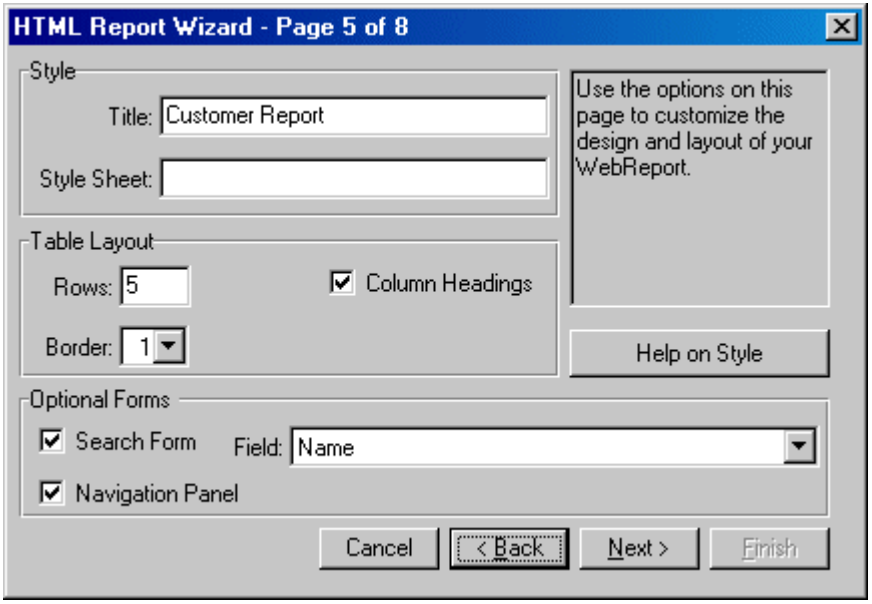


Figure 5–10: Page 5 (Style) Of the HTML Report Wizard

5.3.4 Defining the Links

Follow these steps to define the links from your report to another Web object (that you will create later in this tutorial):

1 ♦ Choose the Next button. Page 6 (HyperLink) appears.

2 ♦ Select Customer.Name from the Field combo box.

3 ♦ In the Web Object field, type **mydetail**. (This is the name of the Web object that you will create later.)

4 ♦ In the Target Frame field, leave the target frame as **_self**.

5 ♦ In the Status Line field, type **Edit customer details**.

6 ♦ Select Rowids in the Parameters section.

When you complete these steps, page 6 should look similar to Figure 5–11.

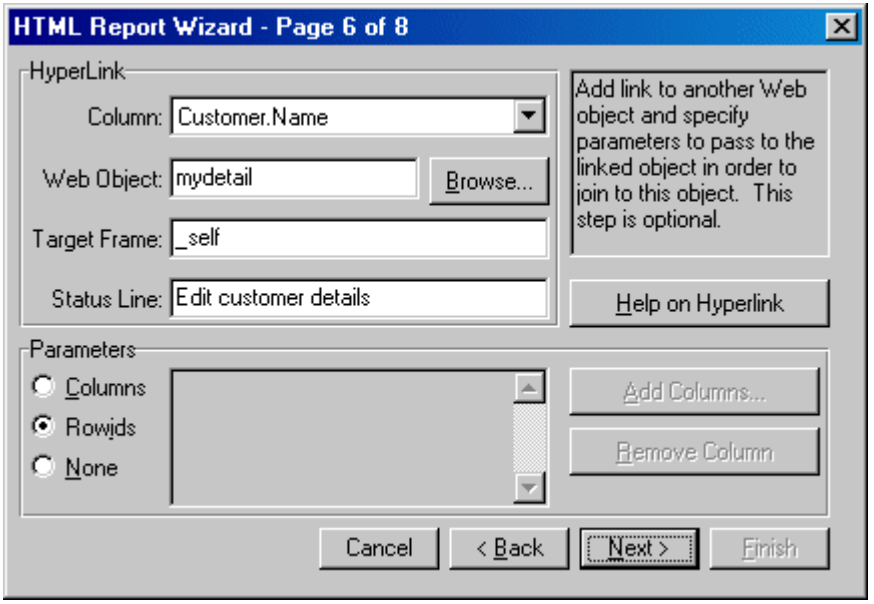


Figure 5–11: Page 6 (Hyperlink) Of the HTML Report Wizard

5.3.5 Defining the Color Of the Report

Follow these steps to define the report's color:

1 ♦ Choose the Next button. Page 7 (Background) appears.

a) In the Page Background color selection table, select a color.

b) In the Table Background color selection table, select a color.

2 ♦ Choose the Next button to continue to Page 8.

5.3.6 Previewing and Saving Your Report

Follow these steps to save the report you have created with the Report Wizard:

1 ♦ Type the filename **mycusrpt.html**, then choose the Preview button. The report is saved and is converted from HTML to a Web object, which is then compiled. A report, similar to Figure 5–12, appears in a browser window:

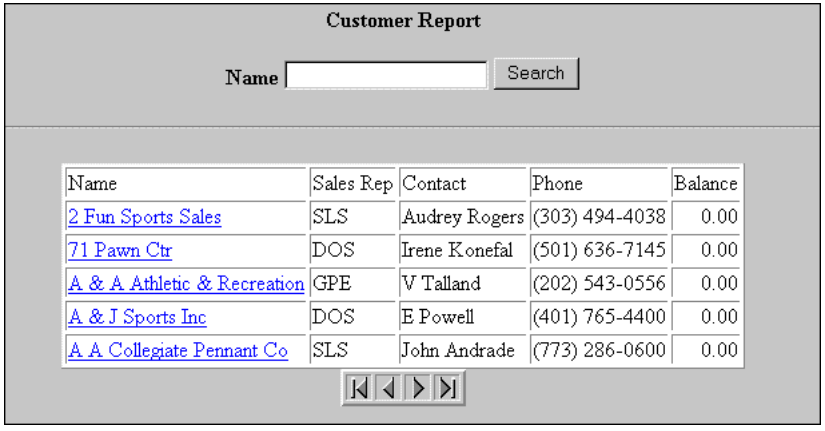


Figure 5–12: Customer Report

NOTE: If you click on a link in the Name column now, you will get an error. You created the links to point to the mydetail Web object, which does not exist yet. You will create mydetail in the next section.

2 ♦ Close the browser window.

3 ♦ Choose the Finish button in the wizard to complete that report. A Procedure Window appears, showing you the HTML source code for the object you have successfully created.

4 ♦ Close the Procedure Window.

5.4 Using the Detail Wizard

The Detail Wizard allows you to create an HTML page that displays an updateable database record.

In this section, you will use the Detail Wizard to create a customer detail update form. When users (sales representatives, for example) choose a customer link from the customer report, they see an individual customer record. They can modify the customer record and return back to the customer report.

5.4.1 Beginning

To create the customer detail update form, you must have a connection to the Sports2000 database. If you do not have a connection, see the "Creating and Connecting To a Database" section in Chapter 3, "Setting Up the WebSpeed Development Environment," for specific instructions.

Follow these steps to launch the Detail Wizard:

1 ♦ From the AppBuilder main window, choose File→ New.

The New dialog box appears.

2 ♦ Choose Detail, and choose OK.

Page 1 of the HTML Detail Wizard appears.

5.4.2 Defining the Query

Follow these steps to define the query:

1 ♦ Choose the Next button to continue to page 2 of the Detail Wizard.

Page 2 allows you to choose whether to base your detail report on an existing SmartDataObject or on a database query.

2 ♦ Choose the Database option. Then choose the Define Query button.

The Query Builder appears.

3 ♦ Select Customer from the Available Tables list, then choose the Add button to move it into the Selected Tables & Joins selection list.

When you complete this step, the Query Builder should look similar to the following:

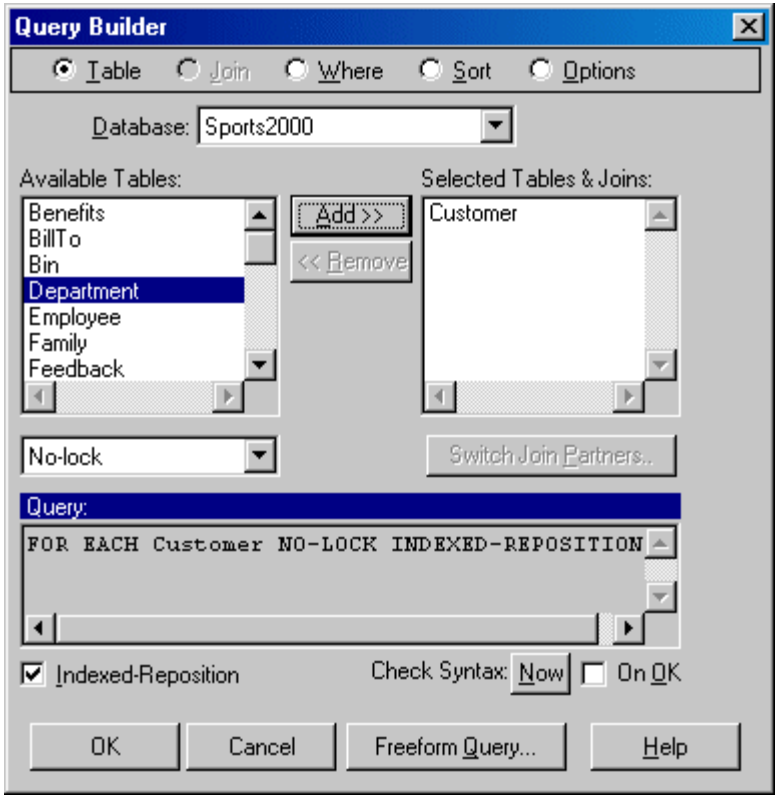


Figure 5–13: Query Builder

4 ♦ Choose OK to close the Query Builder and return to Page 2 of the HTML Detail Wizard.

5 ♦ Choose the Next button.

Page 3, the External Tables page, appears.

5.4.3 Selecting the Fields For the Query

Follow these steps to select the fields for the query:

1 ♦ Choose the Next button to continue. (You will not be selecting any External Tables to complete this form.)

Page 4 (Fields) appears.

2 ♦ Choose the Add Fields button.

The Column Editor dialog box appears.

3 ♦ Choose the Add button.

The Multi-Field Selector dialog box appears.

4 ♦ Select the following fields from the Available Fields selection list: Name, Contact, Balance, Discount, and Terms.

5 ♦ Choose the Add button to move the fields into the Selected Fields list. (You can also double click on the fields to add them to the Selected Fields list.)

In the report, fields appear in the same order as in the Selected Fields list.

6 ♦ Choose OK, then choose OK on the Column Editor dialog box.

Page 4, shown in Figure 5–14, reappears, with the selected fields listed.

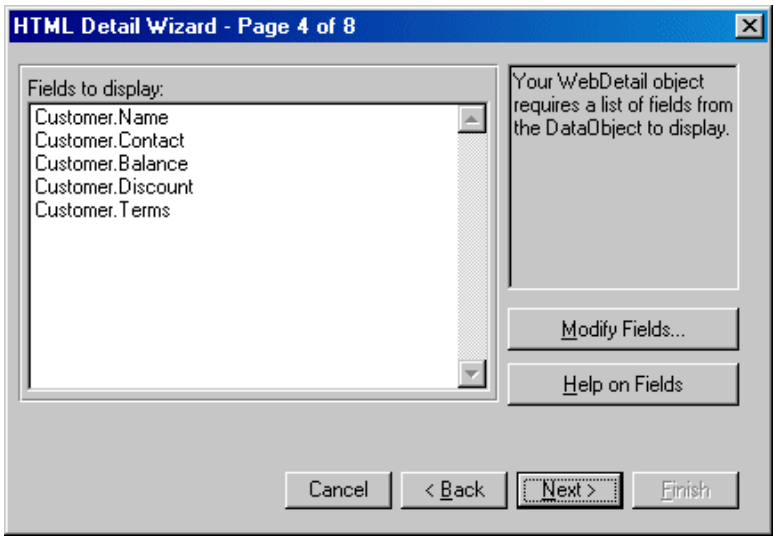


Figure 5–14: Page 4 (Fields) Of the Detail Wizard

5.4.4 Defining the Style Of the Form

Follow these steps to define the form's style:

1 ♦ Choose the Next button on Page 4 of the Detail Wizard.

Page 5, the Style page appears

2 ♦ In the Title field, type **Customer Update**.

3 ♦ In the Update Panel, choose the Add, Reset, Submit, and Delete check boxes.

4 ♦ Deselect the Search Form check box.

5 ♦ Deselect the Navigation Panel check box.

When completed, Page 5 should look similar to the following:

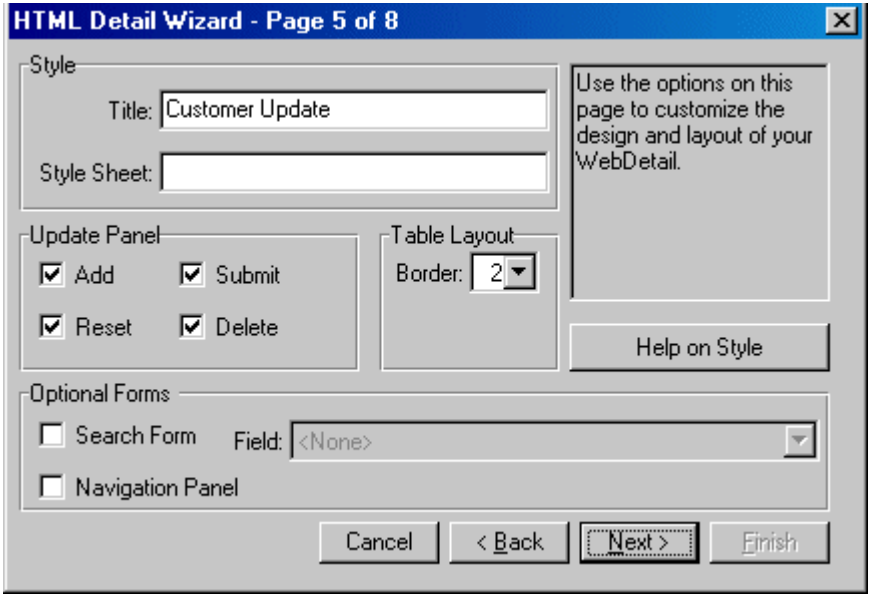


Figure 5–15: Page 5 (Style) Of the Detail Wizard

5.4.5 Defining the Links

Follow these steps to define the links from your detail form back to the customer report you created earlier:

1 ♦ Choose the Next button on Page 5.

Page 6, the HyperLink page appears.

2 ♦ Select Customer.Name from the Column combo box.

3 ♦ In the Web Object field, type **mycusrpt**, which you created earlier.

4 ♦ In the Target Frame field, leave the target frame as _self.

5 ♦ In the Button Label field, type **Back**.

6 ♦ Select None in the Parameters section.

When you complete these steps, page 6 should look similar to Figure 5–16.

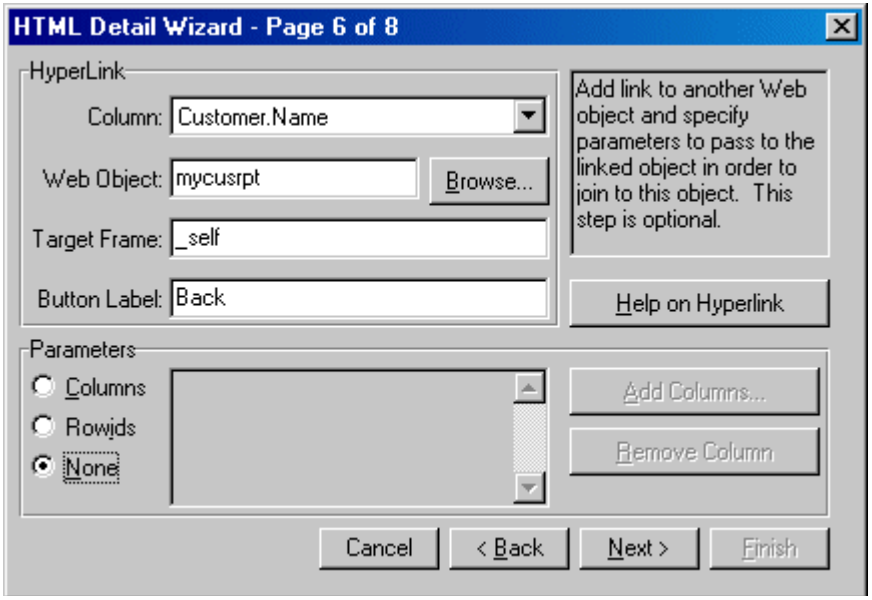


Figure 5–16: Page 6 (Hyperlink) Of the HTML Detail Wizard

5.4.6 Defining the Color Of the Form

Follow these steps to define the form's color:

1 ♦ Choose the Next button on Page 6 of the Detail Wizard.

Page 7, the Backgrounds page, appears.

2 ♦ In the Page Background color selection table, select a color.

3 ♦ In the Table Background color selection table, select a color.

When you complete these steps, Page 7 should look similar to the following:

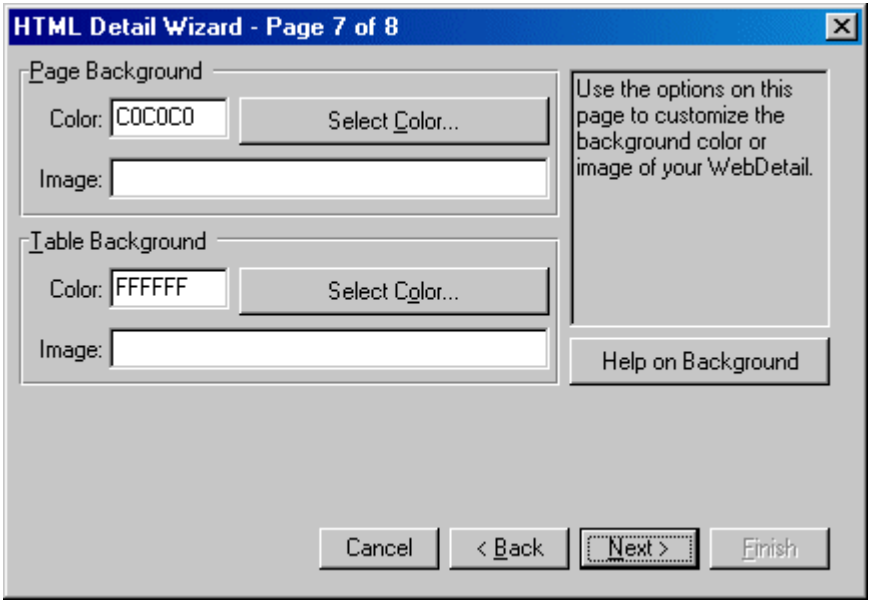


Figure 5–17: Page 7 (Background) Of the HTML Detail Wizard

5.4.7 Previewing and Saving Your Detail Update Form

Follow these steps to save the detail customer update form you created with the Detail Wizard:

1 ♦ Choose the Next button on Page 7 of the Detail Wizard,

Page 8, the final page, appears.

2 ♦ Type the filename **mydetail.html**, then choose the Preview button.

The form is saved and converted from HTML to a Web object, which is then compiled. As shown in Figure 5–18, the report appears in the browser window and opens to the first record.

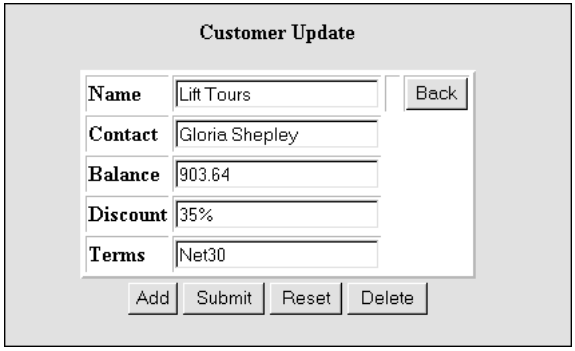


Figure 5–18: Customer Update Form

Choosing the Back button should return you to your Customer Report. Selecting a name in the Customer Report should bring you back the Customer Update form.

The bottom row of buttons should allow you to modify, add, or delete a record.

3 ♦ Close the browser window.

4 ♦ Choose the Finish button in the wizard to complete that form.

5 ♦ Close the Procedure Window that appears.

5.5 Working With Templates

You will use the Main, Blank, and Frameset templates to create the rest of the Web objects for your application.

5.5.1 Creating a Welcome Screen

You are going to use the Main template to create an introductory welcome frame for your application. The Main template is an HTML file with some boilerplate text for a Welcome screen.

Follow these steps to create a Welcome screen:

1 ♦ From the AppBuilder main window, choose File→ New.

The New dialog box appears.

2 ♦ Select Main, and choose OK.

A Procedure Window opens.

3 ♦ Use the Procedure Window scroll bar to scroll through the contents of the Main template.

Notice that the values for host and browser are the agent variables SERVER_NAME and REMOTE_HOST. You can use back tic (‘) escape characters to set off the values of any variables or database fields in an HTML file.

4 ♦ In the Procedure Window, modify the template as shown in the following bold text:

```
<title>Welcome to SportsPro Sales Advisor</title>
</head>
<body bgcolor="white" text="#00008080">
<h1 align="center"><br>
Welcome to SportsPro Sales Advisor<br></h1>
```

5.5.2 Saving Your Welcome Screen

Follow these steps to save your Welcome screen:

1 ♦ Choose File→ Save. The Save As dialog box appears.

2 ♦ In the File name field, type **mymain.html**, then choose the Save button.

3 ♦ Choose Yes for WebSpeed to save and compile the HTML file or report any SpeedScript syntax errors that prevent it from compiling.

5.5.3 Previewing Your Welcome Screen

To preview the Welcome screen you have created with the Main template, choose Compile→ Run from the Procedure Window. As shown in Figure 5–19, a preview of the completed Web object appears in a browser window.

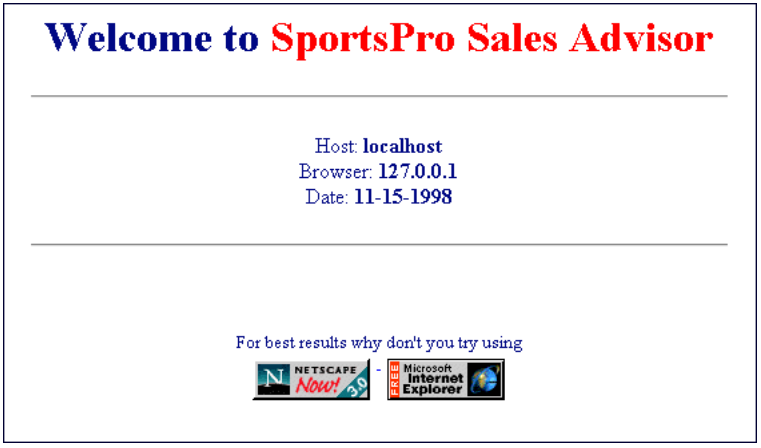


Figure 5–19: Preview Of the Welcome Screen

You can continue to work in the open Procedure Window. When you are satisfied with the results, close the Procedure Window and the browser window in which you have been previewing your work, and begin working on the next Web object.

5.5.4 Creating an Application Header

You are going to use the Blank template to create the Corporate logo Web object. The Blank template creates a simple HTML file with HTML script tags to hold your SpeedScript code.

Follow these steps to create the Corporate logo Web object:

1 ♦ From the AppBuilder main window, choose File→ New.

The New dialog box appears.

2 ♦ Select Blank to create a new file from the Blank template, and choose OK.

A new, untitled file opens in a Procedure Window.

3 ♦ Use the Procedure Window scroll bar to scroll through the contents of the Blank template.

Notice that the template contains standard HTML tags. You can add any standard HTML formatting tags you want. For the Corporate logo Web object you add HTML tags for the background and text colors, logo image, and application title.

Between the embedded scripting language tags, `<script language = “SpeedScript”>` and `</script>`, you add SpeedScript code to define variables to hold the date and time. Outside the script tags, you use backquote escape characters (‘ ‘) to set off the values of each variable.

4 ♦ In the Procedure Window, modify the template by adding the text shown in bold:

```
<body bgcolor=“white” text=“00008080">
<script language=“SpeedScript”>
/*-----------------------------------------------------------------
File:
Description:
Created:
-----------------------------------------------------------------*/
DEFINE VARIABLE vdate AS DATE.
DEFINE VARIABLE vtime AS CHARACTER.
ASSIGN vdate = TODAY.
ASSIGN vtime = STRING(TIME,"HH:MM").
</script>
<img src="/webspeed31d/samples/intranet/corplogo.gif">
<font color="FF0000" size="6" face="Arial">
<strong>Sales Advisor</strong><br>
<p align="center"><font size="3"><strong>Today’s Date: ‘vdate‘ Time:
‘vtime‘</font></p>
</body>
```

NOTE: When you specify an image like `corplogo.gif` in this example, you must specify a path relative to your documentation root directory. The WebSpeed installation creates a directory called `/webspeed`*version* (where *version* is the version number of the WebSpeed product) under your documentation root directory. Various graphics and other static files are installed there.

5.5.5 Saving Your Application Header

Follow these steps to save the Corporate logo you created with the Blank template:

1 ♦ In the Procedure Window, choose File→ Save.

The Save As dialog box opens.

2 ♦ In the File name field, type **mylogo.html**, then choose the Save button.

3 ♦ Choose Yes for WebSpeed to save and compile the HTML file or report any SpeedScript syntax errors that prevent it from compiling.

5.5.6 Previewing Your Application Header

To preview the corporate logo you created with the Blank template, choose Compile→ Run to run the Corporate logo Web object. As shown in Figure 5–20, a preview of the completed Web object runs in the browser window.



Figure 5–20: Preview Of the Application Header

You can continue to work in the open Procedure Window. When you are satisfied with the results, close the Procedure Window and the browser window in which you have been previewing your work, and begin working on the next Web object.

5.5.7 Creating an Index

An Index file is just a static HTML file with links to the Web objects that make up your application. You use the Blank template to create an index for the SportsPro Sales Advisor.

Follow these steps to create the Index Web object:

1 ♦ From the AppBuilder main window, choose File→ New. The New dialog box appears.

2 ♦ Select Blank to create a new file from the Blank template, and choose OK. A new, untitled file opens in the Procedure Window.

3 ♦ In the Procedure Window, modify the template as shown in the following bold text:

```
<body bgcolor="white" text="00008080">
<h1><font face="Arial Rounded MT Bold">Index</h1><br>
<script language="SpeedScript">
/*-----------------------------------------------------
File:
Description:
Created:
-----------------------------------------------------*/
</script>
<p><a href="myinvrpt.html" name="Inventory"
target="SA_main">Inventory Info.</A></p>
<p><a href="mycusrpt.html" name="Customer"
target="SA_main">Customer Info.</A></p>
</body>
```

5.5.8 Saving Your Index

Follow these steps to save the index you created with the Blank template:

1 ♦ Choose File→ Save.

The Save As dialog box opens.

2 ♦ In the File name field, type **myindex.html**, then choose the Save button.

3 ♦ Choose Yes for WebSpeed to save and compile the HTML file or report any SpeedScript syntax errors that prevent it from compiling.

5.5.9 Previewing Your Index

Follow these steps to preview your index:

1 ♦ Choose Compile→ Run.

As shown in Figure 5–21, a preview of the completed Web object runs in the browser window:



Figure 5–21: Preview Of the Index

Test the links. You can continue to work in the open Procedure Window until you are satisfied with the results.

2 ♦ Choose File→ Save to preserve your changes and then close the window.

5.5.10 Creating the Main Sales Advisor Frame Set

You use the Frameset template to create a frame set to contain the Web objects that comprise the SportsPro Sales Advisor application. Follow these steps to create a new frame set for the entire application:

1 ♦ From the AppBuilder main window, choose File→ New.

The New dialog box appears.

2 ♦ Select Frameset to create a new file from the Frameset template, and choose OK.

A new untitled file opens in the Procedure Window.

3 ♦ In the Procedure Window, modify the template as shown in the following bold text:

```
<frameset rows="20%,*">
<frame src="mylogo.html" name="SA_header"
marginwidth="5" marginheight="3">
<frameset cols="20%,*">
<frame src="myindex.html" name="SA_index"
marginwidth="5" marginheight="3">
<frame src="mymain.html" name="SA_main"
marginwidth="5" marginheight="3">
</frameset>
</frameset>
```

5.5.11 Saving Your Frame Set

Follow these steps to save the frame set you created with the Frameset template:

1 ♦ In the Procedure Window, choose File→ Save.

The Save As dialog box opens.

2 ♦ In the File name field, type **myadvise.html**, then choose the Save button.

3 ♦ Choose Yes for WebSpeed to save and compile the HTML file or report any SpeedScript syntax errors that prevent it from compiling.

5.5.12 Previewing Your Frame Set

Choose Compile→ Run in the Procedure Window to preview your frame set.

As shown in Figure 5–22, a preview of the completed Web object runs in the browser window:



Figure 5–22: Preview Of the Frame Set

You see the SportsPro logo in the top frame, the welcome frame in the Main frame of the window, and an index in a narrow frame on the left. When you click on a link in the index, the link opens the option in the Main frame. Test the links and buttons to see how your application functions.

Congratulations! You have just created your first ITP application with the WebSpeed AppBuilder.

5.6 Where To Go From Here

Now that you have successfully created your first WebSpeed application, you have completed this tutorial. To learn more about creating applications with WebSpeed, see the *WebSpeed Developer's Guide*.

Glossary

3GL Programming Language

A general-use programming language such as C, COBOL, FORTRAN, or BASIC that allows the programmer to control a computer at a low level while using a computer-oriented syntax that makes the program code understandable.

4GL Programming Language

An application development language that makes programming much easier because it allows the programmer to use statements that perform the work of multiple 3GL statements and provide intuitive default behavior while using readable English-like syntax.

Agent

See **WebSpeed Agent**

Agent Variables Tool

The WebSpeed WebTool that allows you to display the environment variables for your WebSpeed session.

AppBuilder

The visual programming tool you use to develop applications for client/server or for the Web. For WebSpeed applications, AppBuilder supports creating, testing, and deploying n-tier, HTTP-based applications.

CGI Environment Variables

Standard values, maintained by the HTTP server (part of the Web server), in a named area of shared memory. Common Gateway Interface (CGI) Environment Variables are accessible to other processes running on the Web server, such as the WebSpeed Transaction Messenger, and are often used to pass parameters between the Web server and other applications. The CGI specification defines standard CGI environment variables. In addition, Web servers often maintain their own particular environment variables.

Client

Any application or computer that accesses services on another computer over a computer network.

Database Field

The smallest unit of useful information in a database.

A component of a database record that holds a data value of a single data type. Also called a column.

Databases Tool

The WebSpeed WebTool that allows you to look up information about the database schema (structure) of any database connected to the WebSpeed Agent.

Detail Frame

A frame in the WebSpeed WebTools page that provides an index of links to the various WebTools.

Developer Corner

An information resource for Web programmers who are building ITP applications.

Embedded SpeedScript

SpeedScript code that is embedded in an HTML file within <script> ... </script> HTML tags, or within backquote (‘) escape characters.

Expression Editor

A graphical, calculator-like user interface that is accessible from the Query Builder. A tool for defining expressions that, when evaluated at runtime, determine which records are retrieved from the database.

External Table

A database record retrieved by an object external to the current **Web Object** but accessible to the current Web object.

Extranet

A system of connected Web sites that are separate from the Internet.

File Tools

The WebSpeed WebTool that provides a directory listing from which you can view, open, run, or compile files in the directory, or extract or delete files from the directory.

Form

An interface for capturing information from a user, using text entry fields, option buttons, selection lists graphics, etc.

Frame

1. A SpeedScript container object used to contain the contents of a single **Web Object**. A default frame is defined for each Web object created within the WebSpeed Workshop Environment.

2. An HTML container that displays as a separate scrollable window within a defined **Frame Set**.

Frame Set

A set of HTML frames displayed within a single Web browser window. Each **Frame** in the frame set contains a separate **HTML File** that can be updated or navigated independently of the HTML files contained in the other frames.

Home Page

The World Wide Web site provided by an individual or corporation that presents what is available at or through the site and is considered the primary gateway to that site.

HTML (Hyper Text Markup Language)

A standard text markup language that uses embedded **HTML Tags** to format a file of text. HTML tags allow you to:

- Provide Hypertext links to other documents
- Display graphic images within a document
- Specify typographical effects such as headlines, variable text size, and italics

HTML File

A text file formatted with HTML Tags.

HTML Format Attributes

A set of attributes on the WEB-CONTEXT system handle that specify default **HTML Tags** to format different parts of a SpeedScript **Frame**. You can change these attributes for each Web request. Generally, you change them in pairs that specify a start and an end tag.

HTML Tag

A standard mark-up code in an HTML file that specifies how the Web browser should interpret the file contents. The tag starts with the less than character (<) and ends with the greater than character (>); for example, <P> to mark the start of a paragraph, and </P> to mark the end of the paragraph.

HTML Tag Parser

A program or utility that parses (reads or analyzes) HTML files, locating and extracting information from HTML tags within the file. A Web browser acts as an HTML tag parser, using the information in the HTML tags within each file to determine how to display and format text and images on the screen.

Hyperlink

A navigational link to another document, resource, subject, or site embedded in a graphical or **Hypertext** element.

Hypertext

Any text in an online document that provides a **Hyperlink** to another document, resource, subject, or site. By default, hypertext format is blue, underlined.

Internal Procedure

A code block that you write and compile as part of a larger containing procedure. The block for an internal procedure is defined by the PROCEDURE statement within the containing procedure file.

Internet

An international network of interconnected computers structured to permit organizations and individuals to exchange messages and share information.

Internet Transaction Processing (ITP)

An open framework for developing and deploying Web-based transaction processing applications.

Intranet

A network of interconnected computers, structured to permit individuals within an organization to exchange messages and share information behind the company’s firewall on a LAN or WAN. Traffic is not broadcast across the global Internet.

JAVA

A popular, object-oriented programming language created by Sun Microsystems. JAVA programs, called applets, are stored on a Web server then downloaded and executed by a Web browser. JAVA applets can take advantage of resources on the client computer to create user-interface components beyond the scope of HTML and to allow client-side processing of some user interaction.

JavaScript

A **Scripting Language**, created by Netscape Communications, that is similar to Java in syntax and that can be embedded into an HTML file.

Join

A **Query** operation that requests information from two or more database tables at a time. The output is a results list where each row contains a record from each joined table. Typically, the two queried tables have a common field with a common name; for example, customer.custnum and order.custnum.

Link

See **Hyperlink**

Local Mode

The client/server development mode that allows you to save any files you create to a directory on the client machine and runs any of your objects by a local Progress client.

Main Frame

The work area for the option you select in the Detail frame when working with the WebSpeed WebTools.

Messages Tool

The WebSpeed WebTool that you use to look up the meaning of any WebSpeed error message number.

Natural Join

A join that occurs when two queried tables have a common field with the same field name. For example, a natural join occurs between the CustNum fields of the Customer and Order tables because both tables contain the CustNum field.

Object State Tool

The WebSpeed WebTool that allows you to view the state of procedures running on the WebSpeed Transaction Agent.

Page Component

A SpeedScript element, such as a dynamically generated table, that can be inserted into another Web object. A page component does not generate its own HTTP header.

Parent Browser

The Web browser containing the hyperlink from which the current Web browser was launched.

Persistent Object

A Web Object that resides in the memory of a locked WebSpeed Transaction Agent, available to service requests from a specific Web client (browser). See also **Persistent Procedure, State Persistent**.

Persistent Procedure

An external SpeedScript **Procedure** that you execute with the PERSISTENT option of the RUN statement. After it returns, all data definitions, internal procedures, and user-defined functions contained in the persistent procedure remain available to other SpeedScript procedures and Web objects invoked for the current WebSpeed transaction.

Procedure

Any block of SpeedScript code that can be invoked by name from another SpeedScript procedure using the RUN statement, possibly with run-time parameters.

Procedure File

A source file that contains SpeedScript code that can be compiled into WebSpeed-executable r-code.

ProPath

A WebSpeed environment variable that contains the list of directories a WebSpeed Transaction Agent will search to locate the WebSpeed file it has been requested to run.

Query

A request for data from a database.

Remote Mode

The WebSpeed development mode that allows you to save any files you create to your Web server/Transaction server machine.

Schema

A description of a database’s structure: the tables it contains, the fields within the tables, the indices, and field attributes.

Scripting Lab

The WebSpeed WebTool that allows you to write and edit SpeedScript, SQL, JavaScript, VBScript, or HTML code and run it on the WebSpeed Transaction Agent.

Scripting Language

An interpretive language that supports Web page generation and that generally requires no compilation. Scripting languages are typically limited in scope and provide fewer logic and I/O management features than full programming languages, like SpeedScript and Java.

Server

A process that services requests made by other processes.

A process that access a database in response to client requests.

Server-centric

Primarily residing and executing on the server.

Session Tools

The Databases, Agent Variables, ProPath, and Object State WebSpeed WebTools.

SmartDataObject

A nonvisual SmartObject written in SpeedScript that allows you to define and manage business logic and data verification. The SmartDataObject has at its back end a database query and at its front end an open interface to visual objects.

Source Code

The original version of a procedure file containing SpeedScript code before it is compiled. Source code files usually have the extension .w or .p.

SpeedScript

WebSpeed's native programming language. SpeedScript is an interpreted, block-structured, and statement-oriented 4GL programming language that provides native access to the Progress RDBMS and DataServers, as well as to Web servers for Web page input and output.

SQL (Structured Query Language)

A database language that consists of a set of facilities for defining, manipulating, and controlling data in a relational database. Also called Standard SQL.

State Persistent

Any Web object running on a locked WebSpeed Transaction Agent. When an agent is locked, only one client process can access it.

Tag

See **HTML Tag**

TagExtract

A component of the WebSpeed product. A generic HTML tag parser that parses HTML documents and identifies data-mappable tags. TagExtract is implemented as a DLL that is invoked at design time and at run time to determine data-mappable fields, offsets, field objects, and related utility procedures. An initialization file is used to specify what HTML tags the parser will consider data mappable, what field object types to map them to, and what utility procedures to use.

TagExtract is integral to WebSpeed's ability to map database fields to HTML text and data entry fields.

TCP-IP

See **Transmission Control Protocol/Internet Protocol**

Transmission Control Protocol/Internet Protocol (TCP/IP)

A suite of standard networking protocols that define how computers and applications exchange data over the Internet.

Uniform Resource Locator (URL)

The address of a resource or site on the **World Wide Web**. The URL contains information about the method of access, the server to be accessed, and the path of any file to be accessed. The path for any WebSpeed **Web Object** references the **WebSpeed Messenger**, either directly or indirectly, prior to a reference to the Web object.

Web Browser

The software that allows access to, and formatting and navigation of, information from a server. Popular Web browsers are Microsoft Internet Explorer, NCSA Mosaic, and Netscape Navigator. HTML is key to proper Web-browser functionality.

Web-centric

Focusing on World Wide Web technology.

Web Context ID

The value of the EXCLUSIVE-ID attribute of the WEB-CONTEXT system handle. This is a name/value pair uniquely generated by the **WebSpeed Agent** for each WebSpeed transaction. It contains the name WSEU set equal to a string concatenated from the host name and several integer values that uniquely identify the transaction.

Web Object

An external SpeedScript procedure (a .r file) that can be invoked by a **URL** or executed by another SpeedScript **Procedure** and that generates and services Web pages.

Web Page

An HTML document that is accessible on the Web, typically through a Web browser.

Web Server

Software resident on a machine on the Web to which a Web browser connects. The Web browser requests a file, which the Web server locates and sends to the Web browser.

WebSpeed

Progress Software Corporation's development environment that provides the tools required to create high-performance, scalable, complete Internet Transaction Processing solutions.

WebSpeed Editor

A Workshop editor used to edit code in the current **Procedure File**.

WebSpeed SESSION Attributes

Attributes of the SpeedScript SESSION system handle, showing the session status of the current **WebSpeed Transaction Agent**.

WebSpeed Agent

The WebSpeed application process that executes Web objects, performs database transactions, and dynamically merges data into HTML format. A WebSpeed Agent can be in only one of the following states at any one time:

AVAILABLE—The WebSpeed Transaction Agent is available to receive a Web request.

BUSY—The WebSpeed Transaction Agent is busy processing a Web request.

LIMBO—The WebSpeed Transaction Agent is in limbo because the agent ID has been sent out as available for use.

LOCKED—The WebSpeed Transaction Agent is unavailable because it is currently locked to a particular Web browser.

WebSpeed Broker

The WebSpeed application process that passes Web requests to a **WebSpeed Agent** for execution.

WebSpeed Messenger

The WebSpeed application process that finds an appropriate **WebSpeed Broker** that can process a request from the Web server.

WebSpeed Variables

A set of variables and attributes that shows the transaction state, URL components, and **Web Context ID** of the most recent Web request.

WebSpeed WEB-CONTEXT Attributes

Attributes of the SpeedScript WEB-CONTEXT system handle, showing the **Web Context ID** and **HTML Format Attributes** for the most recent Web request.

WebSpeed WebTools

A collection of utilities that enables developers to easily perform tasks and access information frequently required throughout the development process.

WebSpeed Workshop Environment

The environment in which you develop Web-based Internet Transaction Processing (ITP) applications, which are business applications that support real-time data access and data management. The WebSpeed Workshop Environment includes the AppBuilder, the WebTools, and the WebSpeed Server.

WebStream

A named stream opened with output to the Web from a WebSpeed Agent. All HTML output from Web objects and utility procedures should be put on this stream.

Working Directory

A directory that the installation process automatically identifies and adds to the WebSpeed ProPath. You develop your sample WebSpeed application in your working directory so that WebSpeed can automatically find and run the Web objects and procedure files you create.

World Wide Web

The fastest growing portion of the Internet, the Web uses the HTTP protocol to provide access to a vast array of hyperlinked resources, powerful search capabilities, and multimedia features, including graphics, sound, and video.