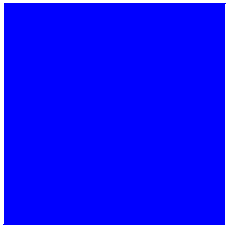


PLATINUM

SQL-Station Coder

User Guide

Version 2.1

PLATINUM
technology

Title and Publication Number

PLATINUM Publication Number: SSC-X-210-UG00-00

Printed: April 7, 1997

Mailing Address

PLATINUM *technology, inc.*
1815 South Meyers Road
Oakbrook Terrace, Illinois
60181-5235

Cover Photo

Photo by Tony Stone/Seattle Images

The animal featured on the cover of this guide is a symbol of PLATINUM's corporate commitment to preserving endangered and threatened species around the globe.

Credits

Writer: Judy Burkhart
Product Team: Jamie Lerner, Dana Heath, Dmitriy Neyman, John Vincent, Robert Gersten, Jeff Burkhart, Bill Wong, Brian Montana, Ariel Zach, Al Garrison, Yuri Zhovnirovsky

Table of Contents

Preface

Philosophy ix
Contacting Technical Support x
About This Guide xi
What's New xiii
Notational Conventions xv
Related Publications xvi

1 • Installation

Overview 1-2
System Requirements 1-2
- System 1-2
- Microsoft Windows Requirements 1-2
- ORACLE Requirements 1-2
- SQL Server Requirements 1-3

Client-Side Installation 1-3
Oracle Server-side Installation 1-5
SQL Server Server-side Installation (Sybase and Microsoft) 1-8

2 • Basics

Overview 2-2
The Coder Main Window 2-4
The Coder Menus and Toolbars 2-5
Getting Started 2-6

Connecting 2-7
The Login Dialog 2-7
Disconnecting 2-9
Groups 2-10
Setting Preferences 2-12
The Catalog Browser 2-12
The Edit Window 2-13
Coder Tools 2-14

3 • Tutorial

Getting Started 3-3
Oracle Tutorial 3-3
Starting Coder and Connecting 3-4
The Edit Window 3-5
Using Object Lookup 3-11
The Catalog Browser 3-12
Generating Table DDL with the SQL Wizard 3-19
Inserting Data Through the Catalog Browser 3-23
SQL Server Tutorial 3-25
Starting Coder and Connecting 3-25
The Edit Window 3-27
Using Object Lookup 3-33
The Catalog Browser 3-35
Generating Table DDL with the SQL Wizard 3-41
Inserting Data Through the Catalog Browser 3-45

4 • The Catalog Browser

Overview 4-3
Getting Started 4-4
The Contents of the Catalog Browser 4-4
Catalog Browser Icons 4-8

Navigating in the Catalog Browser 4-10
Right-click Popup Menu 4-12
Applying Filters 4-13
By Owner 4-14
By Type 4-14
By Name 4-15
Adding and Switching Connections 4-16
Adding Connections 4-16
Switching Connections 4-16
Dropping Objects 4-17
Generating Scripts and Procedures 4-17
Generating a DML Statement for a Table 4-18
Generating a Procedure for a Table 4-18
Generating an Insert Script for a Table 4-19
Reverse-Engineering 4-22
From the Catalog Browser 4-22
Using the SQL Scripting tool 4-22
Viewing and Inserting Data 4-22
Executing Stored Procedures and Functions 4-23

5 • The Edit Window

Overview 5-2
Opening an Edit Window 5-3
Entering and Running SQL Statements 5-4
Setting Edit Window Preferences 5-6
Object Lookup Utility 5-6
Codewright© Features 5-7
Editor Emulation 5-7
Find 5-9
Quick Search 5-9

Version Control 5-10
Syntax Highlighting 5-11
Brace Matching 5-11
Spell Checking 5-12
API Assistance 5-12
Code Templates 5-13

6 • SQL-Station Coder Tools

Overview 6-2
Find Object 6-3
Compare 6-5
Generate SQL Script 6-7
SQL Wizards 6-10
Accessing the SQL Wizards 6-11
Object Lookup Utility 6-12
Table Lookup 6-13
Function Lookup 6-14
Package Lookup 6-15
Procedure Lookup 6-16

7 • Procedure Execution Utility

Overview 7-2
Object Execution Utility 7-2
Procedure Execution Preferences 7-5
SQL Server System SP Execution 7-5

8 • Report Wizard

Overview 8-2
Using the Report Wizard 8-4

9 • Setting Coder Preferences

Setting Coder Preferences 9-2
Catalog Browser Tab 9-3
Edit Window Tab 9-4
Procedure Execution Tab 9-6
General Tab 9-7
Oracle Options tab 9-8
SQL Server Options tab 9-10
Comment Borders tab 9-11
Search and Replace tab 9-12
Debugger tab 9-13

10 • SP Browser

Overview 10-2
Using the SP Browser 10-2

A • Using Coder with Other Tools

Overview A-2
Plan Analyzer for Oracle A-2
Invoking Plan Analyzer for Oracle from Coder A-3
Debugger A-3
Catalog Browser A-4
File Menu A-4

Index

Preface

Philosophy

PLATINUM *technology, inc.*, is the leading vendor of open enterprise systems management (OESM) products, which help organizations manage all the hardware and software components of the multiplatform, multi-operating system, multivendor environment called the open enterprise environment (OEE).

By leveraging its expertise in relational technology, PLATINUM offers products and services that increase the efficiency of individual computing systems and databases, as well as the interoperability of these systems and databases in distributed environments.

Contacting Technical Support

You can contact us with any questions or problems you have. You will be directed to an experienced software engineer familiar with the product in question

For product assistance or information, contact:

USA or Canada, toll free	800-442-6861
Illinois	630-620-5000
FAX	630-691-0708 or 630-691-0406
Internet	info@platinum.com
World Wide Web	http://www.platinum.com

To send Email to PLATINUM Technical Support, use:

Internet	techsup@platinum.com
IBM MAIL Exchange	USRWNPSN

To contact PLATINUM Technical Support, use:

USA or Canada, toll free	800-833-PLAT (7528)
IBM Software Mall	PLATSM4
CompuServe	GO PLATINUM

Our Mailing Address is:

PLATINUM *technology, inc.*
1815 South Meyers Road
Oakbrook Terrace, IL 60181-5235

About This Guide

The *PLATINUM SQL-Station Coder User Guide* explains how to use *PLATINUM SQL-Station Coder* to its fullest capabilities.

This guide assumes that the appropriate *PLATINUM SQL-Station Coder* components have been installed at your site. The instructions for installing the product are in the *Installation* Chapter.

Chapter Number	Chapter Name	Content Description
1	*Installation*	Describes the procedure for installing Coder on the client and server machines.
2	*Basics*	Describes the basic features and functions of Coder.
3	*Tutorial*	Contains a brief tutorial that illustrates some of Coder's features.
4	*The Catalog Browser*	Describes the features of the Catalog Browser, Coder's versatile navigation facility.
5	*The Edit Window*	Describes the features of the Edit Window, Coder's sophisticated editing environment.
6	*SQL-Station Coder Tools*	Describes tools included with Coder, such as the Object Lookup facility and SQL Wizards.
7	*Procedure Execution Utility*	Explains how to use Coder to invoke compiled server-side objects, such as procedures.

Chapter Number	Chapter Name	Content Description
8	*Report Wizard*	Outlines the Report Wizard and available object reports.
9	*Setting Coder Preferences*	Describes Coder's configurable preferences.
10	*SP Browser*	Describes the Coder facility for executing SQL Server system stored procedures.
A	*Using Coder with Other Tools*	Briefly discusses using SQL-Station Plan Analyzer for Oracle and Debugger for Oracle with Coder.
	Index	Helps you locate information within this manual.

What's New

PLATINUM SQL-Station Coder Version 2.1 offers the following new features:

SQL Scripting Facility

Use the Scripting facility to automatically generate a SQL script. Simply highlight all the objects you want to add to a script. SQL-Station will pull in any dependent objects you may have forgotten and order the objects based on dependencies.

Access the Generate SQL Script utility from the **Tools** menu. For information, see the *Generate SQL Script* section of the *SQL-Station Coder Tools* chapter.

Internet Access and Verification of HTML enabled PL/SQL

You can now code and execute Internet-enabled PL/SQL. When executing a stored procedure or package procedure, you can use our Web Execute feature to execute your PL/SQL through your system's default Web Browser.

Access **Web Execute** from the **Procedure Menu** when a Procedure Execution window is open.

Internet Bookmarks

You can specify bookmarks to your favorite web sites. This lets you gather information on the Internet while programming (for example, post questions to technical support, search for the latest database techniques, or download patches and new versions of SQL-Station).

You can set bookmarks from the **Help** menu (**Internet** option).

Code Templates for PL/SQL and Transact-SQL Constructs

Coder provides templates for every standard SQL Server and Oracle programming construct, as well as standard comment templates for all major database objects. You can alter the predefined templates, or add your own.

When an Edit Window is open, access templates from the **Text** menu, or by pressing **Control-t**. For information, see the *Code Templates* section of *The Edit Window* chapter.

Database Object Compare

You can compare any two files or database objects with the new Compare utility (on the **Tools** menu). View all differences side-by-side, displayed with color-coding. For information, see the *Compare* section of the *SQL-Station Coder Tools* chapter.

WHERE Clause Specification in the Generate INSERT Script Utility

The Generate INSERT Script utility now provides for specification of a WHERE clause. This lets you filter the data included in the generated INSERT script.

New Edit Window Features

The Edit Window now supports

- Dynamic syntax highlighting
- Redo (from the **Edit** menu)
- Goto Line (**Control-g**, or **Edit** menu)
- Automatically generated comment blocks (**Text** menu)
- **Control-Tab** to switch between windows

Convert Reports to HTML format

You can now export Coder reports to .html format to be posted on an Intranet or Internet web site. For example, you can post reports of all the tables, procedures, or users in your database to your intranet site so developers can easily find and reuse database objects.

Use the Report Conversion button  on the top of the Report display (final screen of the Report Wizard).

Ability to Invoke SQL Wizards Directly from the Catalog Browser

You can now access the SQL Wizards directly from the Catalog Browser. Simply right-click on an existing object or category, and select **New**.

Notational Conventions

The following notational conventions are used throughout this manual:

- Each manual is divided into chapters. A chapter is a main division that describes a subject matter. Each chapter contains topics that are the major sections of the chapter. Each topic may contain subtopics that further break down the topic.
- The documentation's chapter number and page number appear on each page's footer. The page numbers restart with each chapter.
- The ■ symbol denotes a set of bullet points or instructions.
- NOTE text and WARNING text are designed for easy identification.

Related Publications

As you use this *PLATINUM SQL-Station Coder User Guide*, you might find it helpful to have these additional books available for reference:

- PLATINUM SQL-Station Debugger User Guide
- PLATINUM SQL-Station Plan Analyzer User Guide
- PLATINUM SQL-Station Getting Started Guide
- ORACLE7 Server Concepts Manual
- Codewright for Windows User's Guide

1

Installation

This chapter describes the Installation process for Coder.

Overview **1-2**

System Requirements **1-2**

System 1-2

Microsoft Windows Requirements 1-2

ORACLE Requirements 1-2

SQL Server Requirements 1-3

Client-Side Installation **1-3**

Oracle Server-side Installation **1-5**

SQL Server Server-side Installation (Sybase and Microsoft) **1-8**

Overview

Installing Coder involves establishing a folder on the client (*Client-Side Installation*) and, optionally, installing the Coder views and other structures on a server (*Server-side Installation*). During the installation process, you are prompted to perform a server-side installation. You can choose to perform the server-side install at the same time as the client-side install, or at a later time.

Note • **The Server-side installation must be run on each server you want to access with Coder.**

Coder is installed from the SQL-Station install program. During the installation, you are also prompted to install full or evaluation copies of PLATINUM SQL-Station Debugger and SQL-Station Plan Analyzer. For information on these products, refer to the online or printed User Guides supplied.

System Requirements

Your system must meet the following minimum requirements before you can install and run Coder:

System

- IBM-PC or compatible, 486DX or higher
- At least 8 MB of RAM

Microsoft Windows Requirements

- Windows NT or Windows 95

ORACLE Requirements

- SQLNet 1.0 or higher

- ORACLE 7.1 or higher

SQL Server Requirements

- SYBASE SQL Server 4.x or higher
- Microsoft SQL Server 6.x

Client-Side Installation

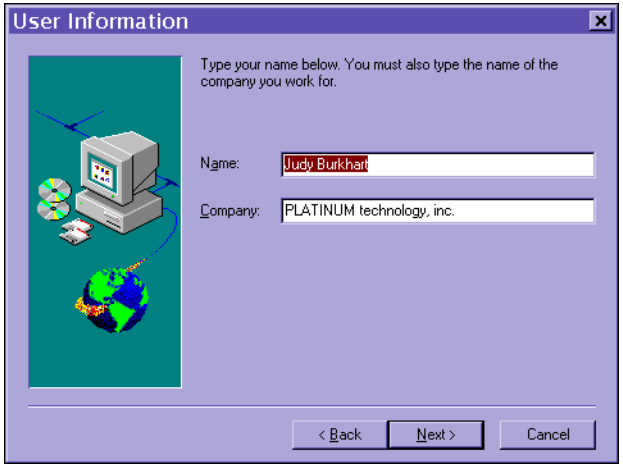
To install Coder on the client machine, follow these steps.

1 Insert the CD-rom in the CD drive.

2 Select **Start, Run...** from the Windows Taskbar (Windows 95 or NT4), or select **File, Run** from Program Manager (Windows NT 3.51).

3 Enter **d:\setup** where **d** is the letter of your CD-ROM.

4 A message box appears indicating the InstallShield wizard is being prepared. After a brief delay, the Welcome dialog appears. Read the information and click **Next** to display the User Information dialog:



Note • At any point during the installation, click **Next** to move forward through the dialogs, or **Back** to return to a prior dialog and revise the information.

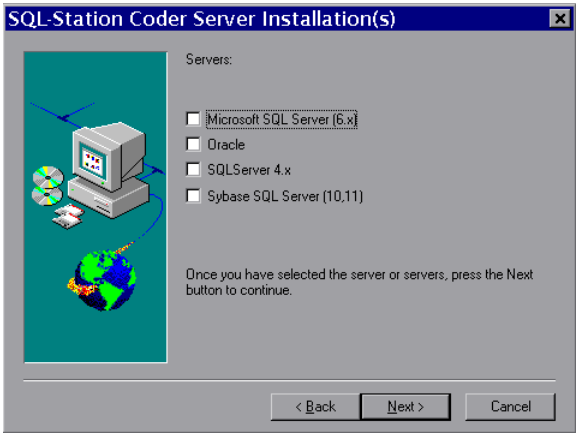
5 Enter your name and company name. Click **Next** to proceed.

6 Follow the subsequent prompts.

Note • **If you choose to install Debugger or Plan Analyzer in addition to Coder, refer to the** *SQL-Station Getting Started Guide* **or the User Guides for each product for information about their installations. These guides are provided on the CD in PDF format, or can be ordered in book form.**

7 When the Server-side Installation dialog appears, enable the **Yes** radio button to perform the server-side install after the Coder files are copied. Enable the **No** radio button to perform only a client-side installation.

8 **Server-***side:* If you have chosen to perform a server-side install, you are prompted to choose the server types you wish to access.



Select the server types to which you want Coder to have access. Click **Next** (or click **Back** to de-select server-side installation). See the next sections on Server Side Installation.

Client-Side Only: If you have chosen only a client-side installation, the Start Copying Files dialog appears, prompting you to verify that the information you have entered is correct. Click **Next** to accept the choices, or **Back** to alter any information you have supplied.

When you have completed this step, Coder copies files into the specified directories.

Oracle Server-side Installation

The server-side installation creates views on the server. You should perform a server-side installation only if the server-side installation has not yet been run on the server you wish to access.

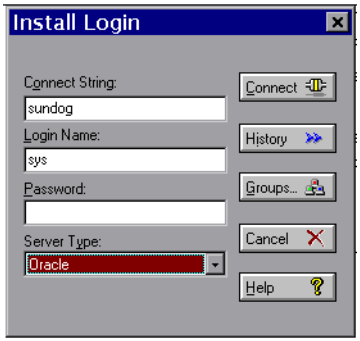
You can choose to perform the server-side installation when you initially install Coder (see previous section), or you can run the server install program at any time after you have performed the client-side installation.

Follow these steps to run the server-side installation after you have already installed Coder:

1 *Windows NT 3.51:* From Program Manager, open the program group into which you installed Coder. Double-Click the icon labeled **Coder Server Install**.

Windows 95, NT 4: From the **Start** Menu, choose **Programs**, and the program folder in which you installed (usually **SQL-Station**). Select the program **Coder Server Install**.

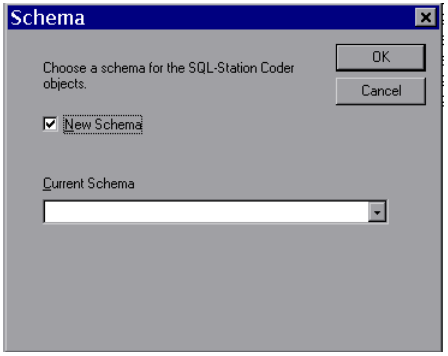
The Install Login dialog appears:



2 Enter the Connect String, Login Name (usually *SYS*), Password, and Server Type (*Oracle*) for the server on which you want to perform the installation.

Note • **To perform the Oracle server-side installation, you must login as SYS or other user with DBA privileges.**

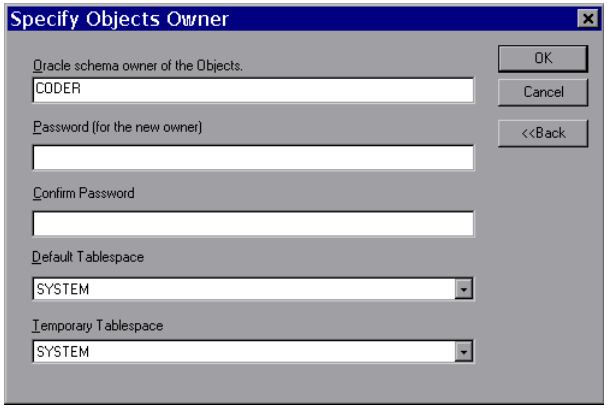
Click **Connect**. The Schema dialog appears:



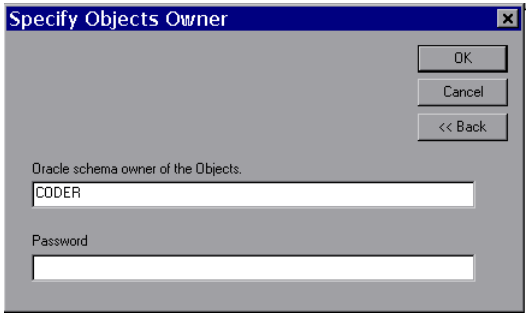
The Oracle server-side installation will install Coder objects on the schema you specify here. Typically, you will want to create a new schema, such as **CODER**.

3 Check the **New Schema** checkbox for a new schema, or choose an existing schema for the Coder objects. Click **OK**.

New Schema: This Specify Objects Owner dialog appears:



Current Schema: This Specify Objects Owner dialog appears:



4 Enter the Schema owner name and password for the current or new owner. If this is a new schema, specify the tablespace information as well. Click **OK**.

The objects are created on the server and the server-side installation finishes.

SQL Server Server-side Installation (Sybase and Microsoft)

The server-side installation creates views on the server. You should perform a server-side installation only if the server-side installation has not yet been run on the server you wish to access.

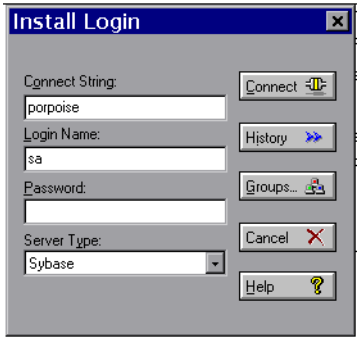
You can choose to perform the server-side installation when you initially install Coder (see previous section), or you can run the server install program at any time after you have performed the client-side installation.

Follow these steps to run the server-side installation after you have already installed Coder:

1 *Windows NT 3.51:* From Program Manager open the program group into which you installed Coder. Double-Click the icon labeled **Coder Server Install**.

Windows 95, NT 4: From the Start Menu, choose **Programs**, and the program folder in which you installed Coder. Select the program **Coder Server Install**.

The Install Login dialog appears:



2 Enter the Connect String, Login Name (*SA*), SA Password, and Server Type for the server on which you want to perform the installation.

Note • **To perform the SQL Server server-side installation, you must login as SA.**

Click **Connect**. The server-side installation status bar displays the progress of the installation. When server-side installation is complete, you can start Coder and connect to the server.

2

Basics

This chapter outlines the basic features of Coder.

Overview **2-2**

The Coder Main Window **2-4**

The Coder Menus and Toolbars **2-5**

Getting Started **2-6**

Connecting **2-7**

The Login Dialog 2-7

Disconnecting 2-9

Groups 2-10

Setting Preferences **2-12**

The Catalog Browser **2-12**

The Edit Window **2-13**

Coder Tools **2-14**

Overview

Coder is an innovative application-development environment in which you can create, view, run, develop, interact with, and maintain database objects. Coder's graphical development environment is easy to use, fast, and flexible. Following are some of the key features of Coder:

- Multiple simultaneous connections to SQL Server and Oracle servers
- Catalog Browser navigation tool
- Code-based scripting facility
- Reverse engineering of all objects, including entire databases or schemas
- Sophisticated Edit window with emulation choices
- System stored procedure execution (SQL Server)
- User Stored Procedure, function, and package execution
- SQL Wizards to facilitate SQL construction
- Report generation wizard
- Integration with other PLATINUM development tools and database tools such as SQL-Station Debugger, SQL-Station Plan Analyzer for ORACLE, CCC/Harvest, and DBVision templates.

Typical Workflow

Interacting with Coder typically involves the following sequence of events:

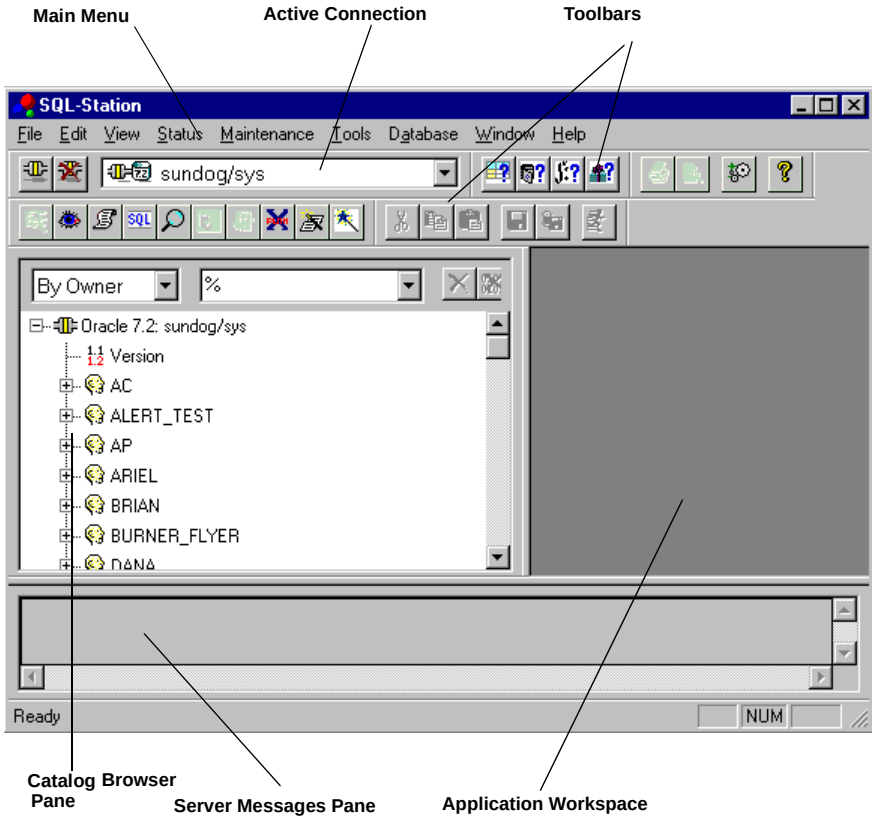
1. Invoke Coder (see *Getting Started*).
2. Connect to a database server (on which the Server-side installation has already been run). (See *Connecting*.)
3. Perform the following operations:

- Use an Edit Window to create, open and edit, run, or view a SQL statement or script. (See *The Edit Window* chapter.)
- Use the Catalog Browser to navigate through and interact with objects on a connected server. Valid interactions include reverse-engineering, executing, and recompiling. (See *The Catalog Browser* chapter.)

4 If you own **SQL-Station Debugger** or **SQL-Station Plan Analyzer**, you can use them within Coder to debug or optimize Oracle objects and statements.

The Coder Main Window

Following is the Coder Main Window:



Note • **Since the toolbars are moveable and configurable, your main window may appear different from the above figure.**

In its default state, the Coder main window includes a Catalog Browser pane, Server Messages Pane, and Application Workspace. You can choose to not display the Catalog Browser or the Server Messages pane. To turn these off, right-click in the pane and choose **Close** or **Hide** from the menu.

The Application Workspace is the area in which Coder displays Edit Windows, Procedure Execution Windows, Debugger Windows, and other dialogs relating to selected objects and actions.

The Coder Menus and Toolbars

The Coder menu options vary slightly depending on which environment you are in: the Catalog Browser or the Edit Window. Additionally, you can access a right-click popup menu that contains context-sensitive options. To access the popup menu, simply click the right mouse button when the cursor is in an Edit Window or Catalog Browser Pane.

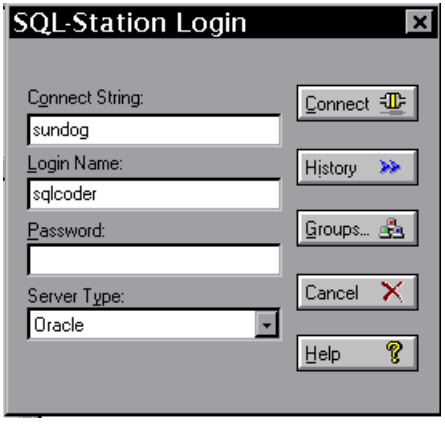
Coder has seven toolbars, each of which can be undocked and toggled on and off:

- Workspace 
- Edit 
- Object Lookup 
- Connection 
- Tools 
- SQL Server Options 
- Oracle Options 

Coder also has a Catalog Browser toolbar that contains the following fields and buttons:

Getting Started

To start Coder, use the **Start** menu (Windows 95, NT4), or double-click the SQL-Station icon from Program Manager (Windows NT 3.51). The SQL-Station main window appears, and the Login dialog comes up.



Click **Cancel** to interact with Coder without connecting, or you can choose to **connect**, list a **history** of connections, or specify or select **groups**.

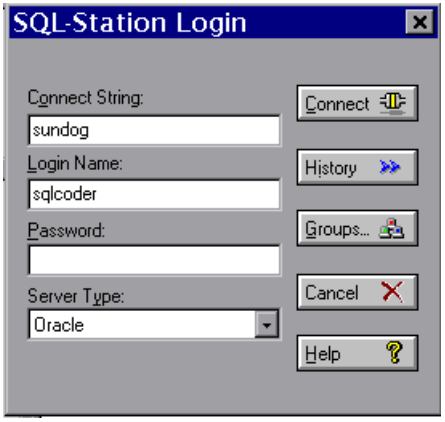
To create or interact with server objects, you must first connect to a server. You can do this when you initially bring up Coder, or at any time thereafter. You can also establish additional connections at any time.

Connecting

Coder's multi-connection capabilities let you maintain connections with one or more servers, logins, and database types.

The Login Dialog

Use the Login dialog to specify, edit, or delete groups, see a history of prior connections, and to establish a connection. The Login dialog appears when you first invoke Coder, or when you select **Connect** from the Coder menu or toolbar.



New Server Connection

Follow these steps to establish a new server connection:

1. If the Login dialog is not already open, select **Database, Open Connection,** or click the connect toolbar button.
2. In the Connect String field, enter a connect string for the database server. Connect strings can be explicit, or can be aliases (if supported by your connectivity software.)
3. In the Login Name field, enter the login name of the user account you want to access.

Note • **The account you specify here will determine which objects you can access through Coder.**

4 Enter the password for the Login name.

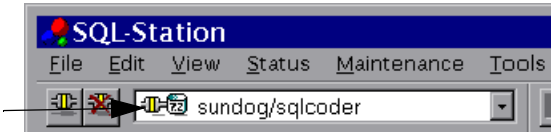
5 In the Server Type field, select the server type to which you want to connect. The combo box lists server types to choose from:

- Oracle
- SQL Server 4.x
- MS SQL Server 6.x
- Sybase
- ODBC

6 Click **Connect** to establish a connection to the selected server

Note • At any point in a Coder session you can select **Database, Open Connection** (or press the toolbar button) to invoke the Login dialog and establish an additional connection.

The active connection displays in the list-box at the top of the Coder main window.

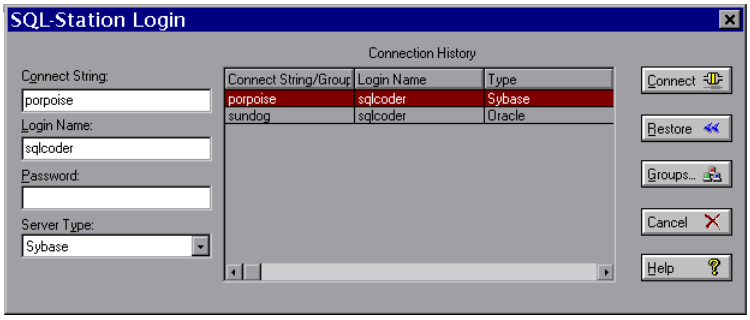


Active Connection List-Box

If you have established multiple connections, you can use the connection list-box to switch between connections. For more information, see the *Adding and Switching Connections* section of *The Catalog Browser* chapter.

Establishing a Prior Connection

From the Login dialog, you can click the **History >>** button to expand the dialog to show a list of all prior connections (including previous Coder sessions). Following is an expanded Login dialog after multiple connections have been made:



Login Dialog Expanded to Show Connection History

Coder keeps track of the last 10 successful connections. You can use the History feature to easily re-establish prior or commonly-accessed connections. Simply double-click the line in the history dialog, or highlight the line and click **Connect**. You are prompted for a password, and the connection is established.

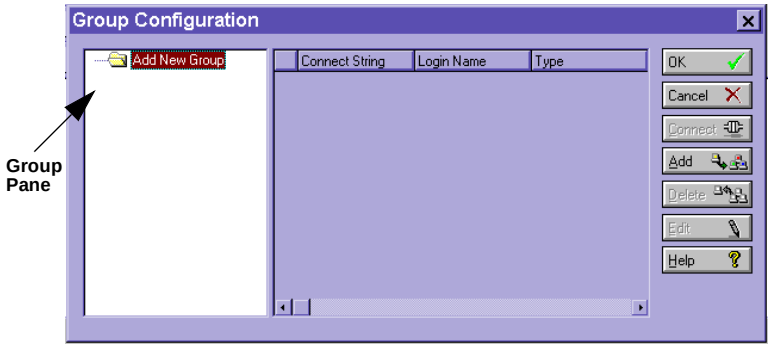
Click **Restore <<** to close the Connection History section.

Disconnecting

To disconnect a connection, select **Database, Disconnect** from the Coder menu, or click the toolbar button. This disconnects the current active connection (the one listed in the Connection list-box).

Groups

From the Login dialog, click the **Groups** button to invoke the Group Configuration dialog. Use the Group Configuration dialog to add, select, remove, or modify Groups of users.



Group Configuration Dialog

Adding a Group

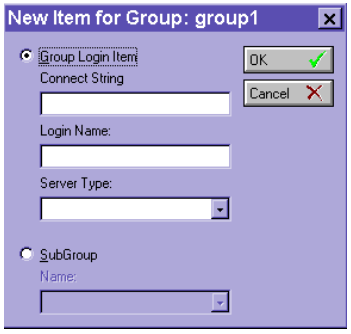
When you first bring up the Group Configuration dialog, the Group pane contains a folder labeled **Add New Group**. Follow these steps to create and populate a group:

1 Highlight the Add New Group folder and click **Add**. The Add New Group dialog appears.

2 Enter a name for the group and click **OK**. A new folder is created in the Group pane with the specified name.

3 To populate the group, highlight the new group folder in the Group pane, and click **Add** (or right-click and select **Add Item** from the popup menu).

The New Item for Group dialog appears:



4 Enter the Connect String, Login Name, and Server Type for a group member, and click **OK**. The member is added to the group.

5 Repeat steps 3 and 4 for each group member you want to add to the group.

Removing Groups or Group Members

At any point you can remove a group or group members by doing one of the following:

- highlight the group name and click **Delete** from the Group Configuration dialog
- right-click the group or group member name and select **Delete** from the popup menu

Setting Preferences

Use the Preferences dialog to set preferences for Coder. The Preferences dialog contains the following tabs:

- Catalog Browser
- Edit Window
- Procedure Execution
- Search and Replace
- Comment Borders
- General
- Oracle Option
- SQL Server Options
- Debugger

For more information see the Chapter *Setting Coder Preferences*.

The Catalog Browser

This section gives an overview of the Catalog Browser. For a more detailed description of the features and functionality of the Catalog Browser, refer to *The Catalog Browser* chapter of this manual.

Use the Catalog Browser to view the contents of a database server, examine the structure, dependencies, and code of the listed objects, and execute certain objects.

You can use the Catalog Browser to do the following:

- Navigate server objects.
- View objects through a variety of filters.

- Reverse-engineer objects to display their creation code.
- Execute stored procedures and functions.
- Recompile server objects.
- Generate Insert Scripts for tables.
- Invoke SQL Wizards to create new objects.
- Insert or modify Table data through the editable Data grid.
- Generate default procedures and packages for tables.
- Move between connections.
- Drop objects.
- View dependencies.
- Compare files.
- View code in a sophisticated editor.
- Invoke the SQL-Station Debugger (if installed on your system).

Use the toolbar, main menu, or right-click popup menu to access these and other Catalog Browser actions.

The Edit Window

This section briefly describes the Edit Window. For a more detailed description of the features and functionality of the Edit Window, refer to *The Edit Window* chapter.

The Edit Window is an environment in which you can edit, run, and create SQL code. You can use the Coder Edit Window to do the following:

- Enter, open, or display SQL code or a script file.
- Execute all or a highlighted portion of the code in the Edit Window.

- Display the generated DDL from a SQL Wizard.
- Open and Edit PL/SQL or Transact-SQL code templates.
- Invoke Plan Analyzer for Oracle.

The Edit Window includes sophisticated editing features of the Codewright© Editor, including the following:

- Syntax Highlighting
- Brace Matching
- File Compare
- Version Control
- Editor Emulation (VI, CUA, Emacs, Brief)
- Search and Replace
- API Assistance

Use the toolbar, main menu, or right-click popup menu to access these and other Edit Window features.

Coder Tools

Coder offers several development tools, all available from the **Tools** menu. Unless indicated otherwise, each tool is documented fully in the *SQL-Station Coder Tools* chapter.

- A Compare utility for comparing database objects or disk files.
- A Debugger for debugging Oracle functions, packages, triggers or procedures (see the *Debugger for Oracle User Guide*).
- The Generate SQL Script utility, which lets you generate the definition SQL for one or more database objects.

- A Find Database Objects utility, for locating database objects on the connected server.
- The Procedure Execution utility, for executing Functions and Procedures without a calling function (see the *Procedure Execution Utility* chapter).
- The SP Browser, which lets you easily select and execute SQL Server administrative procedures (see the *SP Browser* chapter).
- Version Control setup and maintenance facility.
- Codewright's API Assistant Editor.
- SQL Wizards for constructing database objects.
- Report Wizards for a variety of reports (see the *Report Wizard* chapter).
- The Lookup facility, for referencing information about tables, functions, procedures, or packages. You can then drag and drop column names, variable names, statements, or function calls into an Edit Window.
- A Spell Check utility for code in the Edit Window.

3

Tutorial

This chapter provides a walkthrough of a sample Coder session using an Oracle database and a SQL Server database.

Getting Started **3-2**

Oracle Tutorial **3-2**

Starting Coder and Connecting 3-3

The Edit Window 3-4

Using Object Lookup 3-10

The Catalog Browser 3-11

Generating Table DDL with the SQL Wizard 3-18

Inserting Data Through the Catalog Browser 3-22

SQL Server Tutorial **3-24**

Starting Coder and Connecting 3-24

The Edit Window 3-26

Using Object Lookup 3-32

The Catalog Browser 3-34

Generating Table DDL with the SQL Wizard 3-40

Inserting Data Through the Catalog Browser 3-44

Getting Started

Before proceeding with this tutorial, SQL-Station Coder must have been installed on the current machine. Also, the Server-side installation must have been run on at least one database for which you have adequate access privileges. For more information about the server-side installation, see the *Installation* chapter.

Note • **Since this tutorial involves creating a table and procedure, you should not run this tutorial on a Production database.**

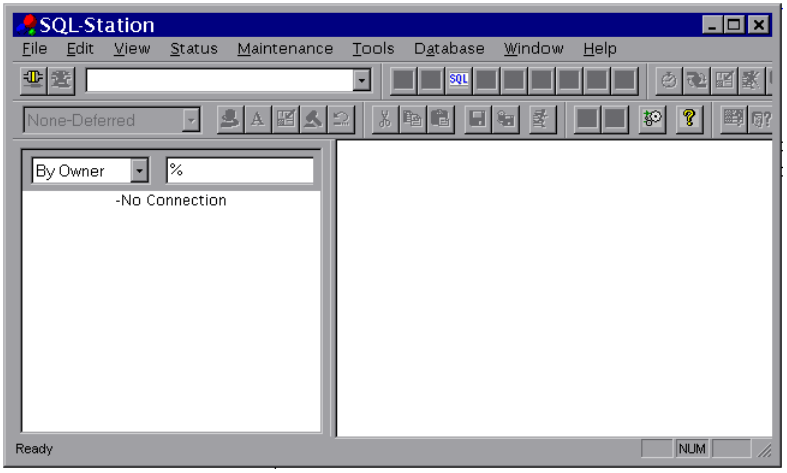
Oracle Tutorial

In this tutorial, you do the following:

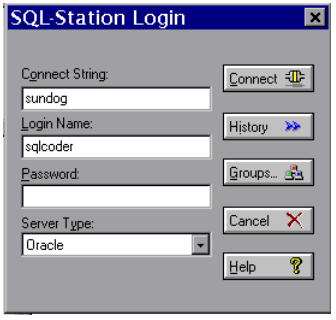
- Start Coder and connect to an Oracle database.
- Use provided SQL scripts to create the following objects:
  - a table (**tutorial_teams**).
  - a stored procedure (**tutorial_get_stats**).
- Insert data into the table using a provided script.
- Enter and execute SQL commands in an Edit Window.
- Look up information about a table.
- Reverse-Engineer an object from the Catalog Browser.
- Execute a stored procedure from the Catalog Browser.
- Generate an Insert Script from the Catalog Browser.
- Use the SQL Wizard to generate DDL for a new table.
- Use the Editable Data Grid to insert data into the new table.

Starting Coder and Connecting

1 Start SQL-Station Coder from the Windows 95 **Start** menu, or from the Windows NT Program Manager. The SQL-Station Main Window appears.



The Login dialog displays:



Note • **If you have started Coder and connected previously, an expanded version of this dialog may appear at this point.**

2 In the Connect String field, type the connect string for a server *on which the Coder Server-side Installation has been run* (for more information, see the *Installation* chapter).

3 In the Login Name field, type a valid login name.

4 In the Password name, type the corresponding password.

5 In the Server Type list-box, select **Oracle**.

6 Click **Connect** to connect to the specified server.

The Coder Main Window displays with the connection listed in the Connection List-Box:

This is the *Active Connection*.

The Active connection is also displayed in the Catalog Browser with this icon:

The Edit Window

1 Open an empty Edit Window by doing one of the following:

- Click the Edit Window toolbar button .
- Select **Tools, Edit Window** from the menu.
- Select **File, New** from the menu.

An Edit Window opens in the Application Workspace.

Note • **The Edit Window is associated with the active connection at the time it is opened. Any commands you run from this Edit Window are executed on the associated connection.**

Note • **You can also open an Edit Window by opening an existing .sql file.**

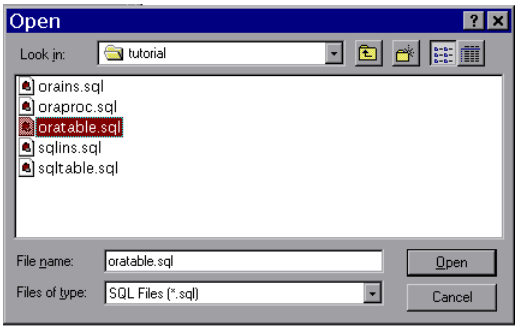
2 Select **File, Close Edit** to close the Edit Window.

Running a Create Table Script

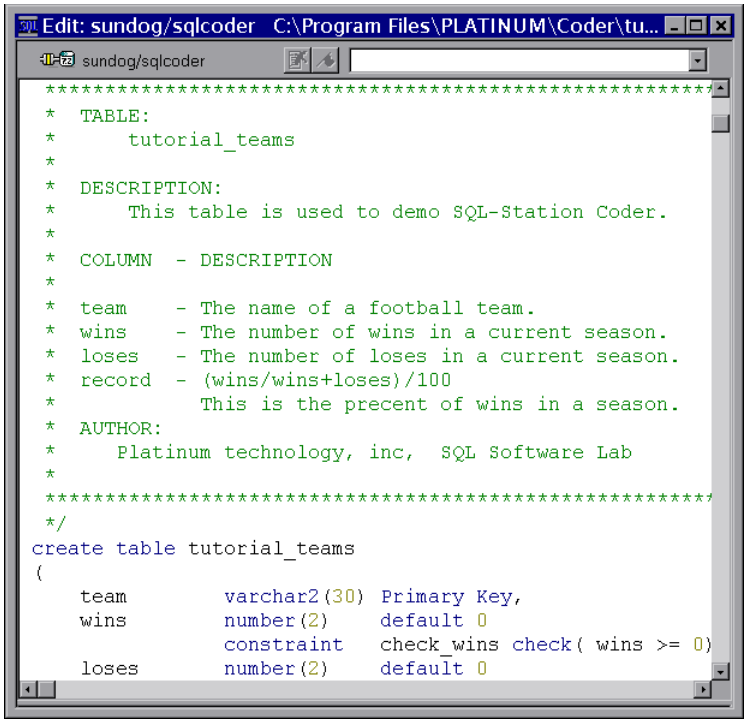
To create a table on the Active Connection, follow these steps:

1 Select **File, Open** from the menu.

2 Navigate to the **Tutorial** directory under SQL-Station Coder. Select the **oratable.sql** file.



A new Edit Window displays the contents of **oratable.sql**:



The statements in this file creates a table called **tutorial_teams**.

3 Click the Execute toolbar button to run the statement.

The statement executes and the table is created. A results tab appears in the lower portion of the Edit Window. The Results tab displays the statement that was executed. The Status bar at the bottom of the SQL-Station window displays a Command Successful message.

You have now created the **tutorial_teams** table.

4 Select **File, Close Edit** to close the Edit Window.

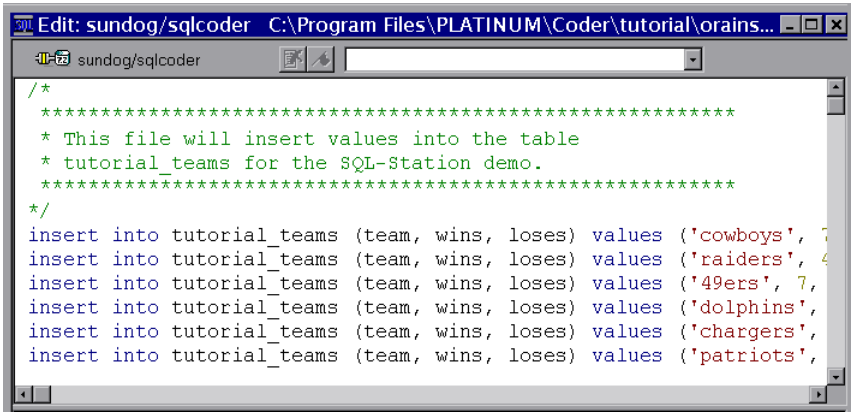
Running an Insert Script

Follow these steps to insert data into the **tutorial_teams** table:

1 Select **File, Open**.

2 Navigate to the **Tutorial** directory under SQL-Station Coder. Select the **orains.sql** file.

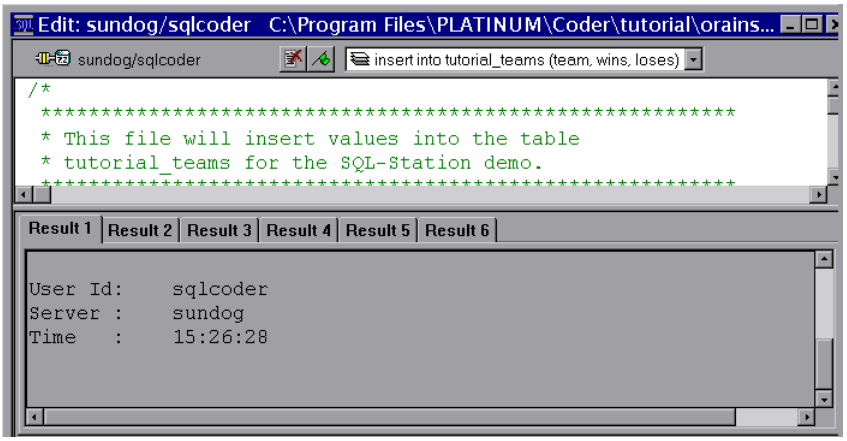
A new Edit Window displays the contents of **orains.sql**:



The statements in this file insert data into the **tutorial_teams** table created in the previous section.

3 Click the Execute toolbar button  to run the statement.

The statement executes and the data is inserted. The Edit Window displays a Result Tab for each statement.



4 Select **File, Close Edit** from the menu to commit the transaction and close the Edit window.

Note • **If you do not close the Edit Window at this point, you can click the Commit button on the toolbar to commit the transaction.**

Entering and Running a New SQL Statement

Now type a SELECT statement to verify that the table and data exist on the server:

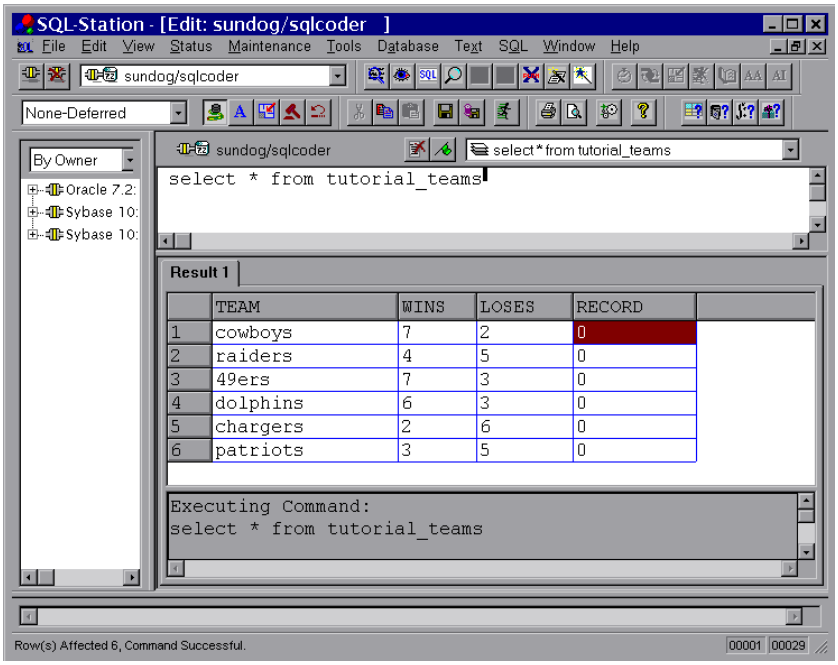
1 Select **File, New** or click the Edit Window toolbar button to open an empty Edit Window.

2 Type the following in the Edit Window:

```
Select * from tutorial_teams
```

3 Click the Execute toolbar button.

The statement runs, and results are displayed in the Results Pane.



4 Select **File, Close Edit** to close the Edit Window. Click **No** to the Save prompt.

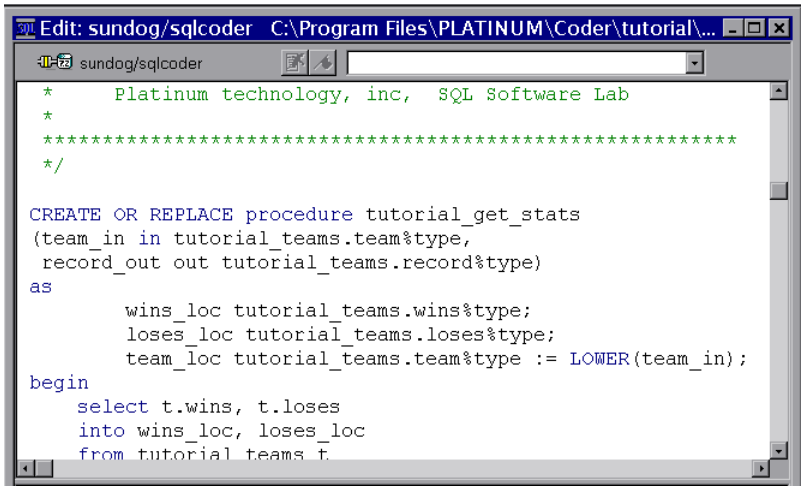
Running a Create Procedure Script

Follow these steps to create a Stored Procedure for this tutorial.

1 Select **File, Open** from the menu.

2 Navigate to the **tutorial** subdirectory and select the file **oraproc.sql**.

A new Edit Window opens and displays the contents of **oraproc.sql**:



The statements in this file create a procedure called **tutorial_get_stats**. This procedure takes an input parameter (team name) and outputs the seasonal percentage of wins for that team.

3 Click the Execute toolbar button to run the statement.

The statement executes and the procedure is created. The Edit Window displays a Result Tab showing the statement that was executed.

Later in this tutorial, you will execute this procedure.

Using Object Lookup

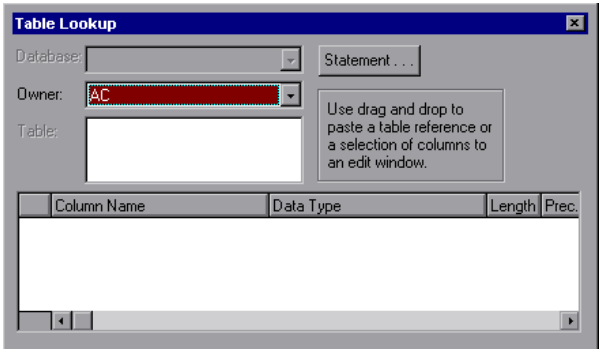
The Object Lookup feature lets you easily look up information about a table, procedure, package, or function. In this section, you will look up information about the table you created

You can access Object Lookup from the toolbar or from the **Tools** menu.

Lookup Table

1 Click the Lookup Table toolbar button (or select **Tools, Lookup, Table** from the menu).

The Table Lookup dialog appears:



2 In the Owner field, select the account under which you created the **tutorial_teams** table.

3 In the Table field, select **tutorial_teams**.

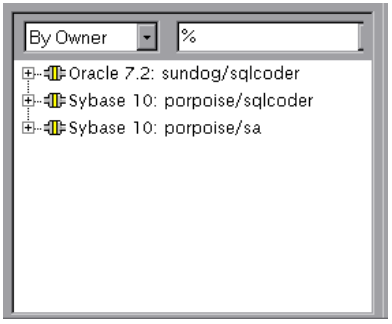
You can now drag the table or selected columns into an Edit Window for use in a SQL statement.

You can also click the **Statement** button to generate the DDL for a SQL statement based on the selected table. This statement can be dragged and dropped into an Edit Window, and then executed.

The Catalog Browser

The Catalog Browser is a sophisticated environment for navigating through and interacting with server objects. (For detailed information on Catalog Browser features, see*The Catalog Browser* chapter.)

The Catalog Browser displays a connection icon for each Connection.



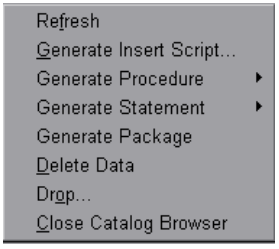
Double-click a connection icon to expand it. The connection icon expands to show User Accounts . You can continue expanding the tree to display the objects for each User Account, and then relevant actions for each object.

- Navigate through the Catalog Browser by double-clicking the icons.

Right-click Popup Menu

Coder includes a context-sensitive right-click popup menu. From within the Catalog Browser or Edit Window, you can click the right mouse-button to see a menu of available actions. Within the Catalog Browser, the contents of the popup menu vary depending on the type of object or icon selected.

Following is the right-click popup menu when a table object is highlighted:



Reverse-Engineering

Reverse-Engineering means generating the DDL code for an existing server object. Follow these steps to reverse-engineer the **tutorial_get_stats procedure**:

1 In the Catalog Browser, navigate to the User Account on which you created the procedure (your current login).

2 Double-click the User Account to display the object icons.

3 Scroll down to the Procedures category, and double-click it to display individual procedures.

4 Locate **tutorial_get_stats** and double-click it.

5 Click the Code icon .

Coder opens a new Edit Window with the code for the **tutorial_get_stats** procedure.

6 Close the Edit Window, click **No** when prompted to save.

Note • **You could also use the Generate SQL Script utility from the Tools menu to reverse-engineer one or more objects.**

Executing a Stored Procedure

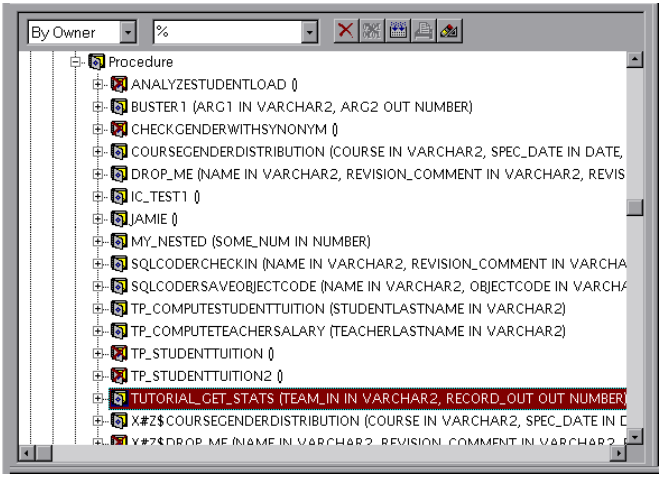
Coder includes a mechanism for executing a Stored Procedure without having to write a calling function. This *Procedure Execution* facility is accessible from the Catalog Browser (when a stored procedure is highlighted) or from the toolbar or **Tools** menu.

1 In the Catalog Browser pane, navigate to the User Account on which you created the **tutorial_get_stats** procedure.

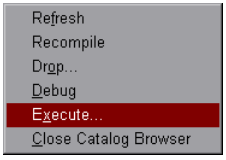
2 Expand the User Account to display the object categories.

3 Scroll down to the Procedures category and double-click it to display the procedures.

The procedure names are displayed with the names of input and output parameters (in parenthesis).

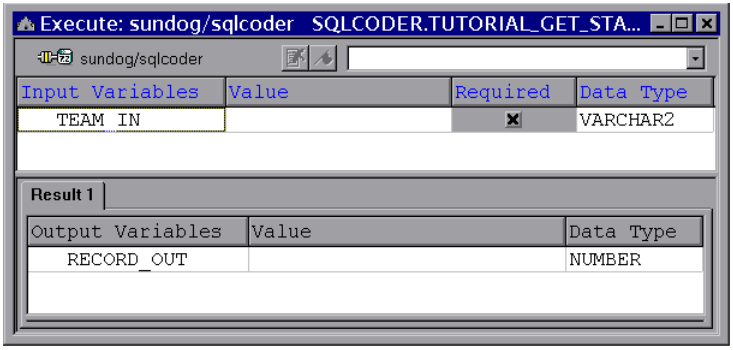


4 Highlight the **tutorial_get_stats** procedure name, and click the right mouse-button. The Right-click popup menu appears:



Select **Execute** from the menu.

A Procedure Execution window appears in the Application Workspace:



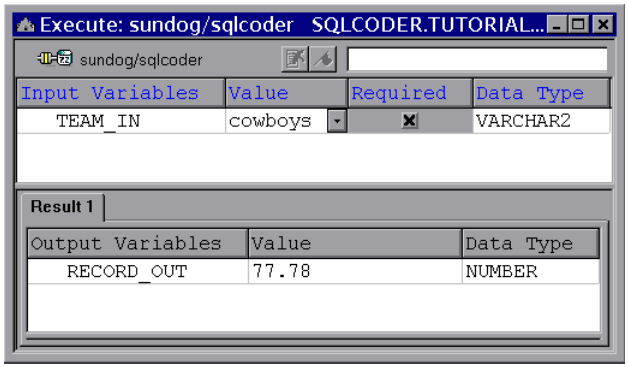
The Procedure Execution window contains a section for specifying input parameters.

5 The **tutorial_get_stats** procedure requires a **team_in** input parameter. Enter the value **cowboys** in the Value field.

This specifies we want to find statistics for the Cowboys football team.

6 Click the Execute toolbar button to execute the procedure.

Coder executes the procedure, and displays the output value in the Value field of the Results pane. In this way, you can execute a stored procedure multiple times, specifying different input values, and comparing the output.



7 Repeat steps 5 and 6 to execute the procedure again, but this time specify **49ers** for the **team_in** parameter.

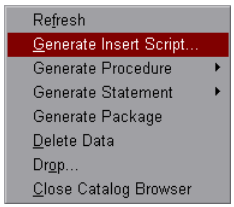
Generating an Insert Script

From within the Catalog Browser, you can easily generate an Insert script for any table. To generate an Insert script, you simply highlight the tablename and select **Generate Insert Script** from the right-click popup menu. Coder then analyzes the table structure and its data, and generates a SQL script. In this way, you can easily duplicate table structure or data between identical or modified versions of tables.

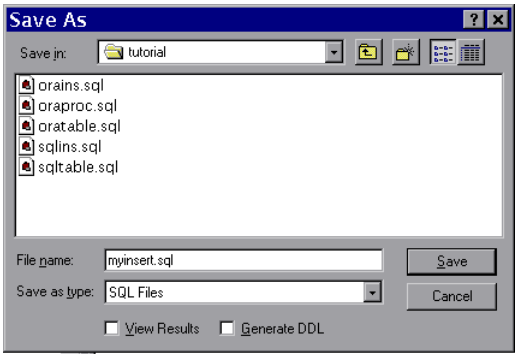
Follow these steps to generate an Insert script for the **tutorial_teams** table:

1 In the Catalog Browser Pane, navigate to the User Account that contains the **tutorial_teams** table.

2 Highlight the table name and click the right mouse-button. The right-click popup menu appears:



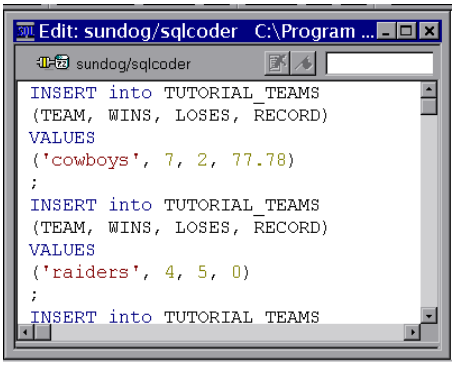
3 Select **Generate Insert Script**. The Save As dialog appears:



Specify **myinsert.sql** as the name in which to save the Insert Script, and click **Save**.

Coder displays a message box stating that the generation was successful.

4 You can open the **myinsert.sql** file and view the output:



Coder generated an Insert statement for every row of existing data in the **tutorial_teams** table.

You could use this script to add new rows to the table, simply by editing the values in each insert statement.

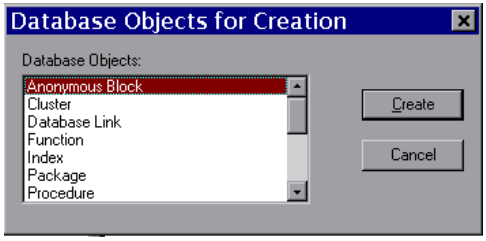
Generating Table DDL with the SQL Wizard

You can use SQL Wizards to easily generate DDL for a variety of objects, such as Triggers, Procedures, or Indexes. The SQL Wizards provide a series of dialogs, checkboxes, and field-names in which you specify criteria. With a simple button-click, Coder generates the code needed to create the object.

To create a new table with the SQL Wizard, follow these steps:

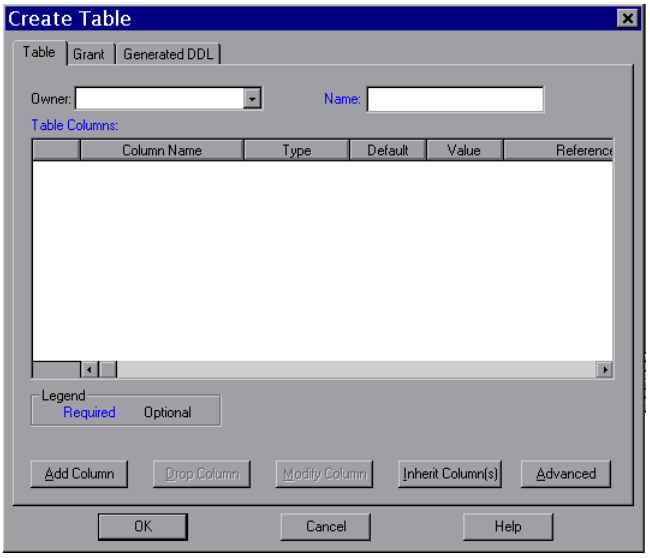
1 Click the SQL Wizard toolbar button, or select **Tools, SQL Wizards** from the menu.

The Database Objects for Creation dialog appears:



Scroll down the list of object types to **Table**. Highlight Table, and click **Create**.

The Create Table Wizard appears:



The Create Table Wizard lets you create a simple or complex table. The required fields are labeled in blue. For example, the simplest table you can create must contain values for the **Name** and **Table Columns** fields; so in the above dialog, those fields are colored blue.

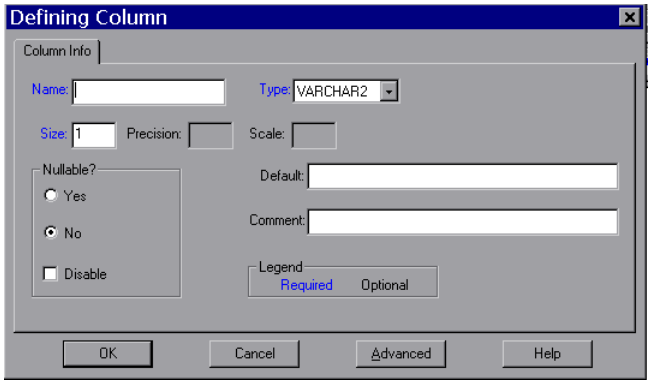
For this exercise, you will create a simple table.

2 In the Name field, type **emp**.

Note • **If the existing account already has a table named *emp*, specify a different name. You will need to substitute this different name for *emp* throughout this tutorial.**

3 To add a column, click the **Add Column** button.

The Defining Columns dialog appears:



4 Type **LNAME** in the Name field.

5 In the Type field, select **VARCHAR2**.

6 In the Size field, type **15**.

7 Click **OK** to add the column.

8 Click **Add Column** to add a second column.

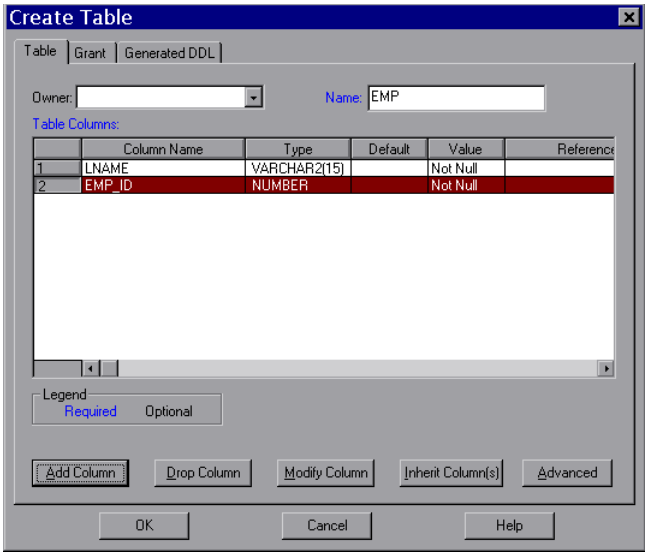
The Defining Column dialog displays again.

9 In the Name field, type **EMP_ID**.

10 In the Type field, choose **NUMBER**.

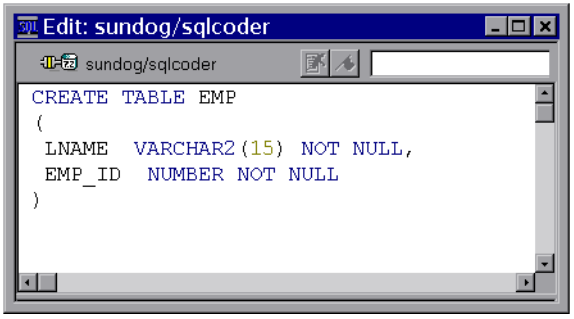
11 Click **OK** to add the emp_id field.

The Add Table wizard should now look as follows:



12 Click the Generated DDL tab to view the DDL. Notice that the statements are based on the information you have entered in the previous steps.

13 Click **OK** to transfer the DDL code to a new Edit Window.



You can now execute, modify, or save the DDL to a file.

14 To execute the CREATE TABLE statement, verify the new Edit Window has focus and click the Execute toolbar button . Coder executes the statement and creates the **emp** table.

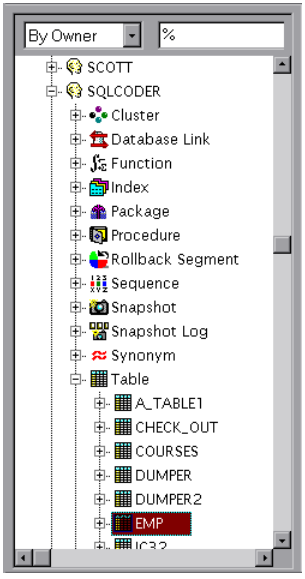
Inserting Data Through the Catalog Browser

There are multiple ways to insert data into a table. In this section, you use the Coder editable grid to insert data into the **emp** table.

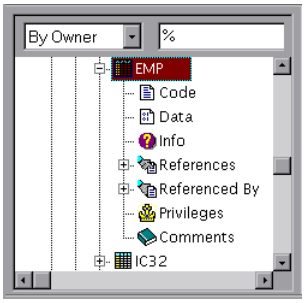
1 Switch to the Catalog Browser, and click the right mouse-button to display the menu.

2 Select **Refresh** to view the server in its current state.

3 Navigate through the Catalog Browser to find the **emp** table you created in the previous section: double-click the account name, and double-click the tables icon.



4 Double-click **emp** to display the associated actions:



5 Double-click the Data icon to display a data grid in the Application Workspace.

The grid appears with [NULL] values in the fields.

6 Highlight the first cell (in the LNAME field) and type **Lerner**.

Coder inserts a new row in the grid.

7 Highlight the cell under EMP_ID and type the value **001**.

8 To specify another row to insert, click in the [NULL] field under LNAME.

9 Type **Heath** and **002** in the LNAME and EMP_ID fields, and press the <Return> key.

Coder does not actually insert the data until you specify to commit the transaction, or you close the Edit Window.

10 Select **File, Close Data** to close the Edit Window and commit the transaction.

11 Verify the data was inserted by doing one of the following:

- Click the Data icon under the **emp** table.
- Open a new Edit Window, type the statement **Select * from emp**, and click the Execute button .

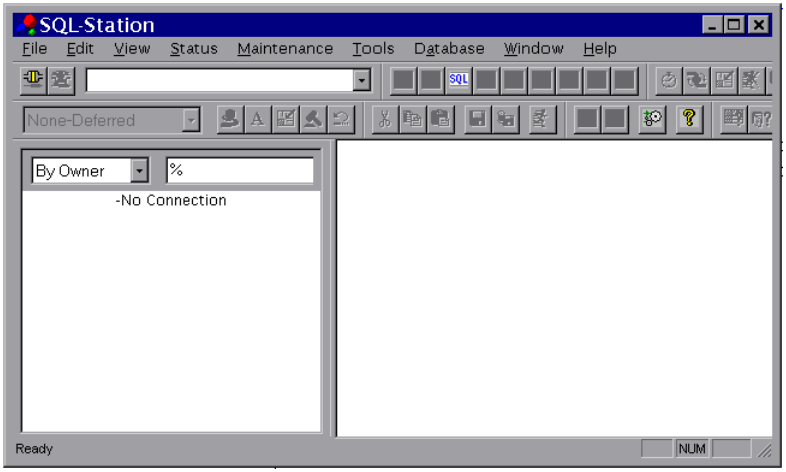
SQL Server Tutorial

In this tutorial, you do the following:

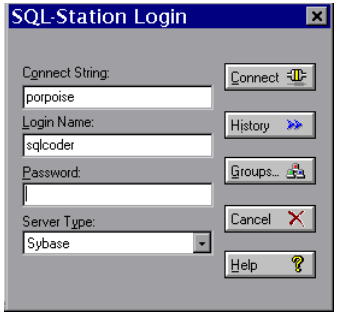
- Start Coder and connect to a SQL Server database.
- Use provided SQL scripts to create the following objects:
  - a table (**tutorial_teams**).
  - a stored procedure (**tutorial_get_stats**).
- Insert data into the table using a provided script.
- Enter and execute SQL commands in an Edit Window.
- Look up information about a table.
- Reverse-Engineer an object from the Catalog Browser.
- Execute a stored procedure from the Catalog Browser.
- Generate an Insert Script from the Catalog Browser.
- Use the SQL Wizard to generate DDL for a new table.
- Use the Editable Data Grid to insert data into the new table.

Starting Coder and Connecting

1 Start SQL-Station Coder from the Windows 95 **Start** menu, or from the Windows NT Program Manager. The SQL-Station Main Window appears.



The Login dialog displays:



Note • **If you have started Coder and connected previously, an expanded version of this dialog may appear at this point.**

2 In the Connect String field, type the connect string for a server *on which the Coder Server-side Installation has been run* (for more information, see the *Installation* chapter).

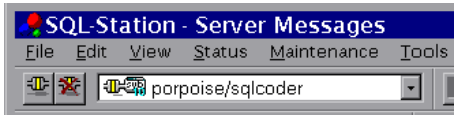
3 In the Login Name field, type a valid login name.

4 In the Password name, type the corresponding password.

5 In the Server Type list-box, select **Sybase** or **SQL Server**.

6 Click **Connect** to connect to the specified server.

The Coder Main Window displays with the connection listed in the Connection List-Box:



This is the *Active Connection*.

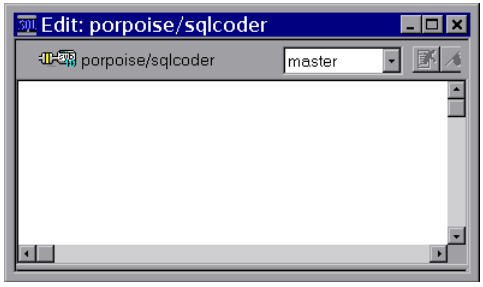
The Active connection is also displayed in the Catalog Browser with this icon:

The Edit Window

1 Open an empty Edit Window by doing one of the following:

- Click the Edit Window toolbar button .
- Select **Tools, Edit Window** from the menu.
- Select **File, New** from the menu.

An Edit Window opens in the Application Workspace.



The Edit Window is associated with the active connection at the time it is opened. Any commands you run from this Edit Window are executed on the associated connection.

Note • **You can also open an Edit Window by opening an existing .sql file.**

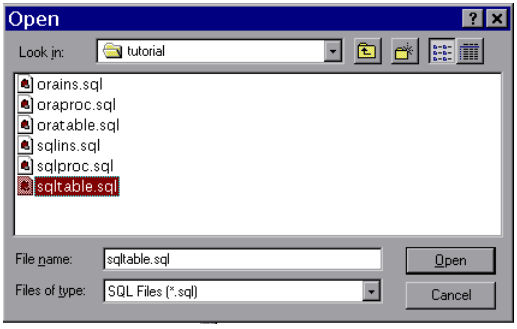
2 Select **File, Close Edit** to close the Edit Window.

Running a Create Table Script

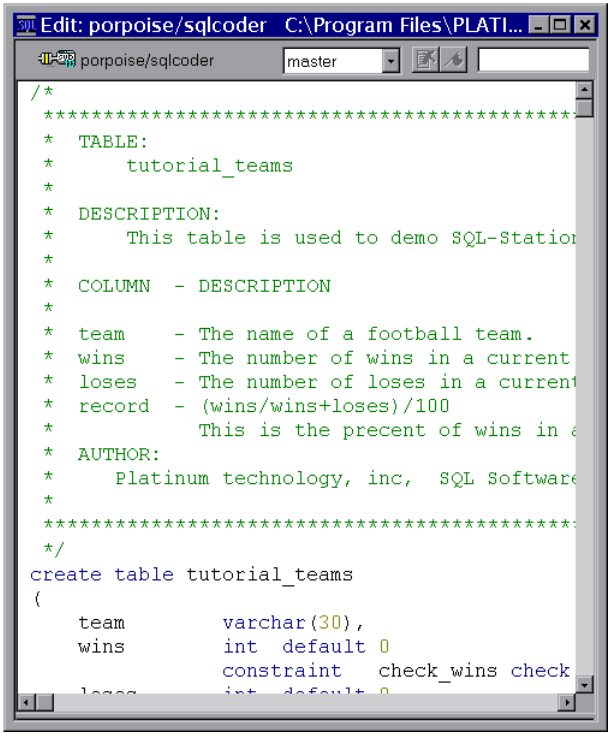
To create a table on the Active Connection, follow these steps:

1 Select **File, Open** from the menu.

2 Navigate to the **Tutorial** directory under SQL-Station Coder. Select the **sqltable.sql** file.



A new Edit Window displays the contents of **sqltable.sql**:



The statements in this file create a table called **tutorial_teams**.

3 Click the Execute toolbar button to run the statement.

Note • **You may need to change the database in the Database listbox at the top of the Edit Window before executing the statement.**

The statement executes and the table is created. A results tab appears in the lower portion of the Edit Window. The Results tab displays the statement that was executed. The Status bar at the bottom of the SQL-Station window displays a Command Successful message.

You have now created the **tutorial_teams** table.

4 Select **File, Close Edit** to close the Edit Window.

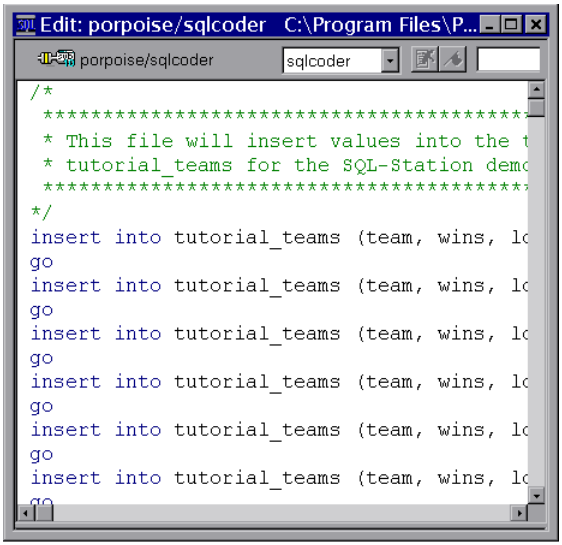
Running an Insert Script

Follow these steps to insert data into the **tutorial_teams** table:

1 Select **File, Open**.

2 Navigate to the **Tutorial** directory under SQL-Station Coder. Select the **sqlins.sql** file.

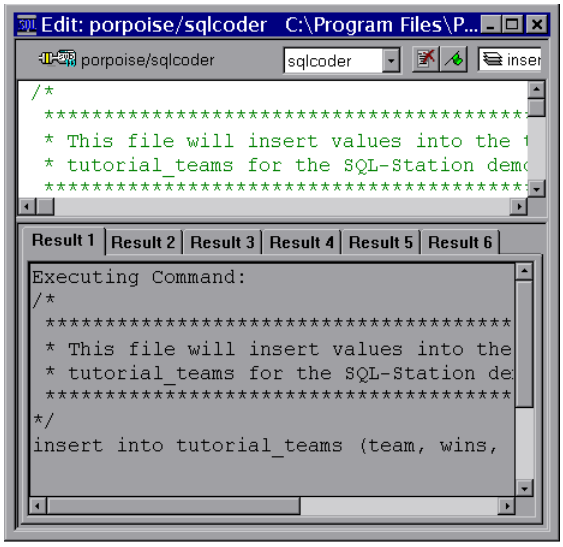
A new Edit Window displays the contents of **sqlins.sql**:



The statements in this file insert data into the **tutorial_teams** table created in the previous section.

3 Click the Execute toolbar button to run the statement.

The statement executes and the data is inserted. The Edit Window displays a Result Tab for each statement.



4 Select **File, Close Edit** to close the Edit window.

Entering and Running a New SQL Statement

Now type a SELECT statement to verify that the table and data exist on the server.

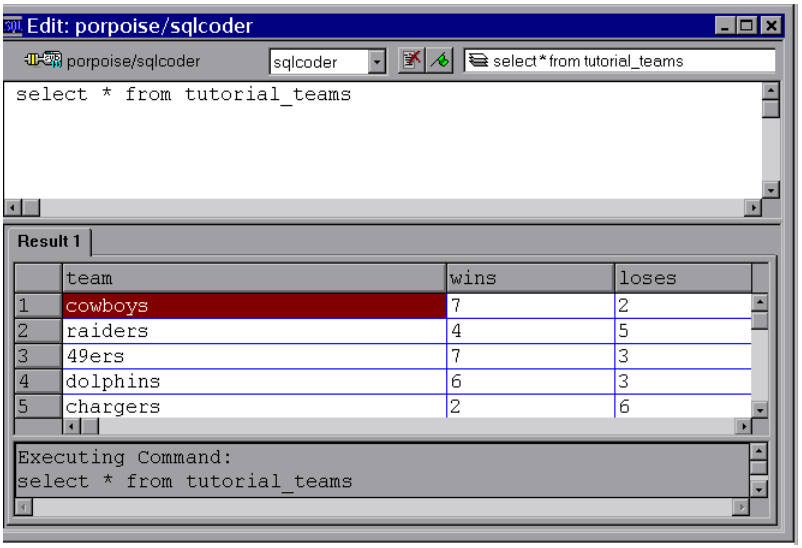
1 Select **File, New** or click the Edit Window toolbar button to open an empty Edit Window.

2 Type the following in the Edit Window:

Select * from tutorial_teams

3 Click the Execute toolbar button.

The statement runs, and results are displayed in the Results Pane.



4 Select **File, Close Edit** to close the Edit Window. Click **No** to the Save prompt.

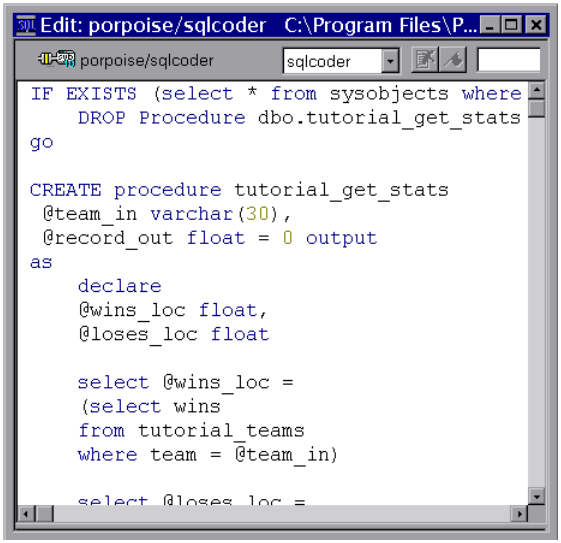
Running a Create Procedure Script

Follow these steps to create a Stored Procedure for this tutorial.

1 Select **File, Open** from the menu.

2 Navigate to the **tutorial** subdirectory and select the file **sqlproc.sql**.

A new Edit Window opens and displays the contents of **sqlproc.sql**:



The statements in this file create a procedure called **tutorial_get_stats**. This procedure takes an input parameter (team name) and outputs the seasonal percentage of wins for that team.

3 Click the Execute toolbar button to run the statement.

The statement executes and the procedure is created. The Edit Window displays a Result Tab showing the statement that was executed.

Later in this tutorial, you will execute this procedure.

Using Object Lookup

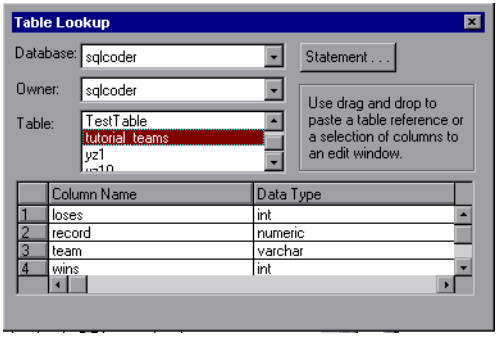
The Object Lookup feature lets you easily look up information about a table, procedure, package, or function. In this section, you will look up information about the table you created.

You can access Object Lookup from the toolbar or from the **Tools** menu.

Lookup Table

1 Click the Lookup Table toolbar button (or select **Tools, Lookup, Table** from the menu).

The Table Lookup dialog appears:



2 In the Database field, select the database under which you created the **tutorial_teams** table.

3 In the Owner field, select the account under which you created the **tutorial_teams** table.

4 In the Table field, select **tutorial_teams**.

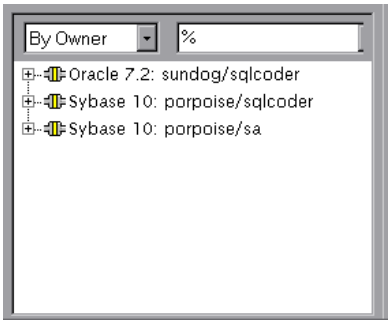
You can now drag the table or selected columns into an Edit Window for use in a SQL statement.

You can also click the **Statement** button to generate the DDL for a SQL statement based on the selected table. You can then drag and drop the SQL statement to an Edit Window to execute it.

The Catalog Browser

The Catalog Browser is a sophisticated environment for navigating through and interacting with server objects. (For detailed information on Catalog Browser features, see *The Catalog Browser* chapter.)

The Catalog Browser displays a connection icon for each Connection.



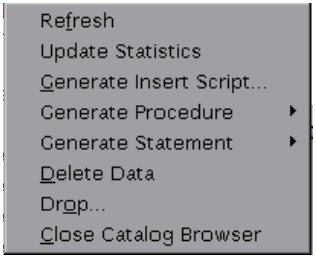
Double-click a SQL Server connection icon to expand it. The connection icon expands to show Databases. Databases expand to show User Accounts. You can continue expanding the tree to display the objects for each User Account, and then relevant actions or components for each object.

Navigate through the Catalog Browser by double-clicking the icons.

Right-click Popup Menu

Coder includes a context-sensitive right-click popup menu. From within the Catalog Browser or Edit Window, you can click the right mouse-button to see a menu of available actions. Within the Catalog Browser, the contents of the popup menu vary depending on the type of object or icon selected.

Following is the right-click popup menu when a User Table object is highlighted:



Reverse-Engineering

Reverse-Engineering means generating the DDL code for an existing server object (or entire database). Follow these steps to reverse-engineer the **tutorial_get_stats procedure**:

1. In the Catalog Browser, navigate to the Database and User Account on which you created the procedure (your current login).
2. Double-click the User Account to display the object icons.
3. Scroll down to the Procedures category, and double-click it to display individual procedures.
4. Locate **tutorial_get_stats** and double-click it.
5. Click the Code icon .

   Coder opens a new Edit Window with the code for the **tutorial_get_stats** procedure.
6. Close the Edit Window, click **No** when prompted to save.

Executing a Stored Procedure

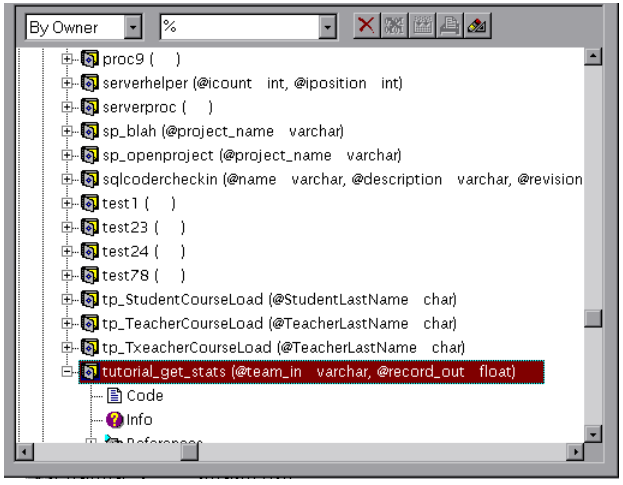
Coder includes a mechanism for invoking a Stored Procedure without having to write a calling function. This *Procedure Execution* facility is accessible from the Catalog Browser (when a stored procedure is highlighted) or from the toolbar and **Tools** menu.

1 In the Catalog Browser pane, navigate to the User Account on which you created the **tutorial_get_stats** procedure.

2 Expand the User Account to display the object categories.

3 Scroll down to the Procedures category and double-click it to display the procedures.

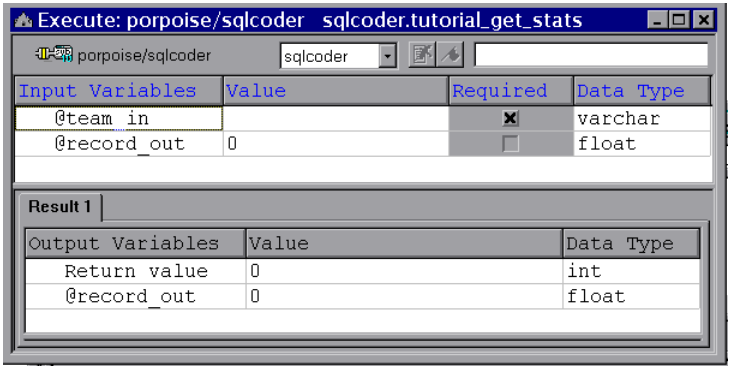
The procedure names are displayed with the names of input and output parameters (in parenthesis).



4 Highlight the **tutorial_get_stats** procedure name, and click the right mouse-button. The Right-click popup menu appears:

Select **Execute** from the menu.

A Procedure Execution window appears in the Application Workspace:



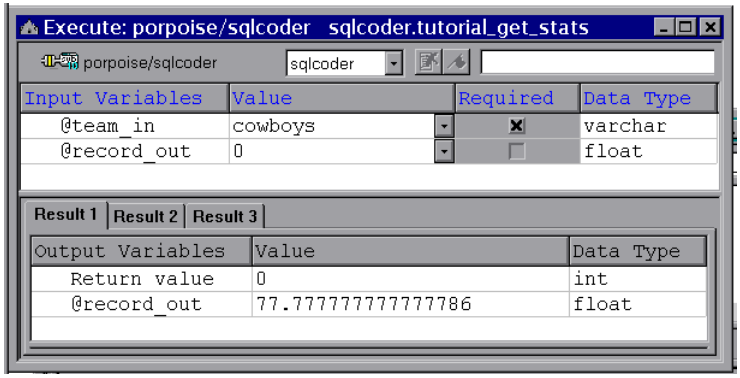
The Procedure Execution window contains a section for specifying input parameters.

5 The **tutorial_get_stats** procedure requires a **team_in** input parameter. Enter the value **cowboys** in the Value field.

This specifies we want to find statistics for the Cowboys football team.

6 Click the Execute toolbar button to execute the procedure.

Coder executes the procedure, and displays the output value in the Value field of the Results pane. In this way, you can execute a stored procedure multiple times, specifying different input values, and comparing the output.



7 Repeat steps 5 and 6 to execute the procedure again, but this time specify **49ers** for the **team_in** parameter.

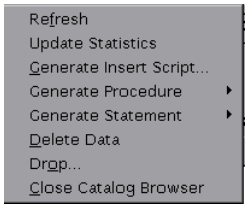
Generating an Insert Script

From within the Catalog Browser, you can easily generate an Insert script for any table. To generate an Insert script, you simply highlight the tablename and select **Generate Insert Script** from the right-click popup menu. Coder then analyzes the table structure and its data, and generates a SQL script. In this way, you can easily duplicate table structure or data between identical or modified versions of tables.

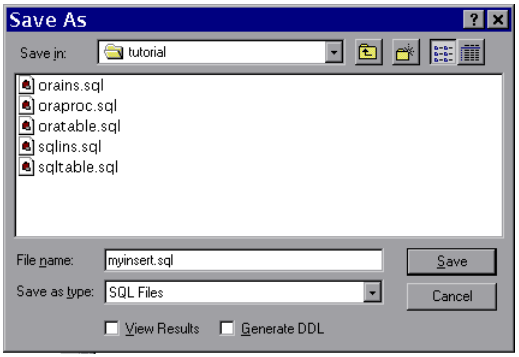
Follow these steps to generate an Insert script for the **tutorial_teams** table:

1 In the Catalog Browser Pane, navigate to the User Account that contains the **tutorial_teams** table.

2 Highlight the table name and click the right mouse-button. The right-click popup menu appears:



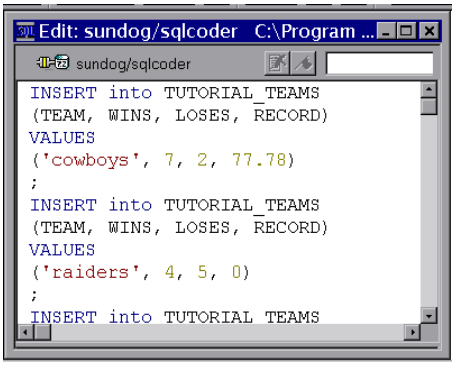
3 Select **Generate Insert Script**. The Save As dialog appears:



Specify **myins.sql** as the name in which to save the Insert Script, and click **Save**.

Coder displays a message box stating that the generation was successful.

4 You can open the **myins.sql** file and view the output:



Coder generated an Insert statement for every row of existing data in the **tutorial_teams** table.

You could use this script to add new rows to the table, simply by editing the values in each insert statement.

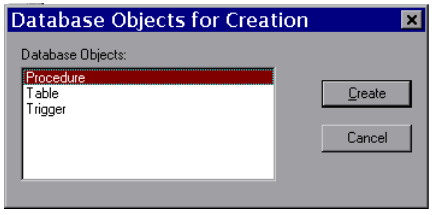
Generating Table DDL with the SQL Wizard

You can use SQL Wizards to easily generate DDL for Triggers, Procedures, or Tables. The SQL Wizards provide a series of dialogs, checkboxes, and field-names in which you specify criteria. With a simple button-click, Coder generates the code needed to create the object.

To create a new table with the SQL Wizard, follow these steps:

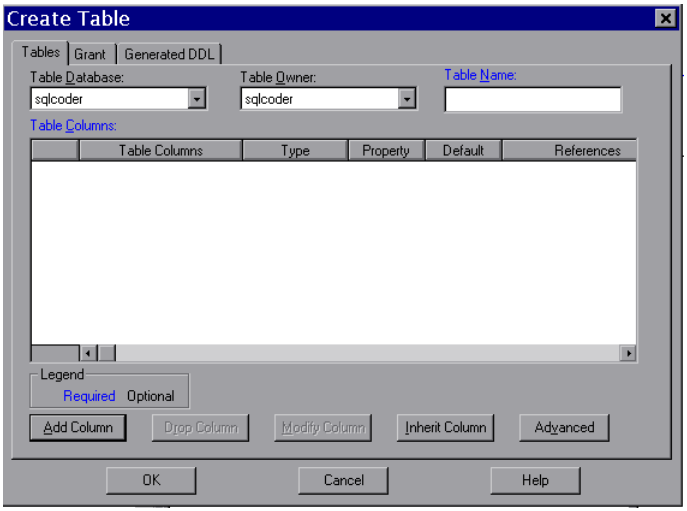
1 Click the SQL Wizard toolbar button, or select **Tools, SQL Wizard** from the menu.

The Database Objects for Creation dialog appears:



Scroll down the list of object types to **Table**. Highlight Table, and click **Create**.

The Create Table Wizard appears:



The Create Table Wizard lets you create a simple or complex table. The required fields are labeled in blue. For example, the simplest table you can create must contain values for the **Table Name** and **Table Columns** fields; so in the above dialog, those fields are colored blue.

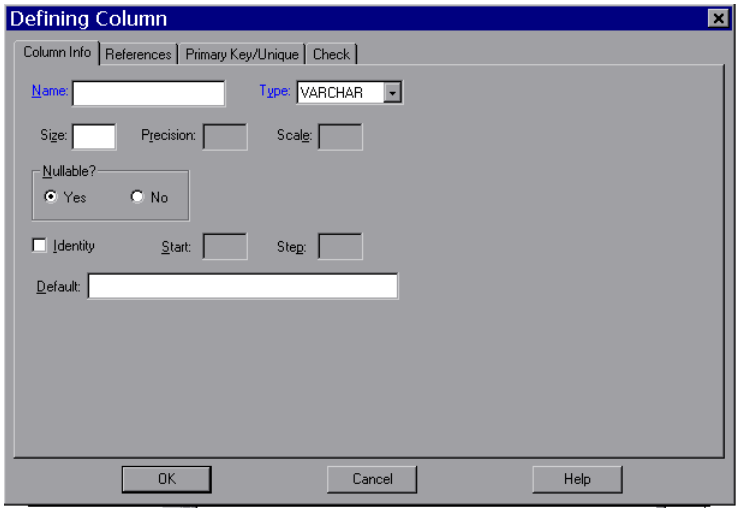
For this exercise, you will create a simple table.

2 In the Name field, type **emp**.

Note • **If the account you are working in already contains a table named *emp*, specify a different name. You will need to substitute this different name for *emp* throughout this tutorial.**

3 To add a column, click the **Add Column** button.

The Defining Columns dialog appears:



4 Type **LNAME** in the Name field.

5 In the Type field, select **VARCHAR**.

6 In the Size field, type **15**.

7 Click **OK** to add the column.

8 Click **Add Column** to add a second column.

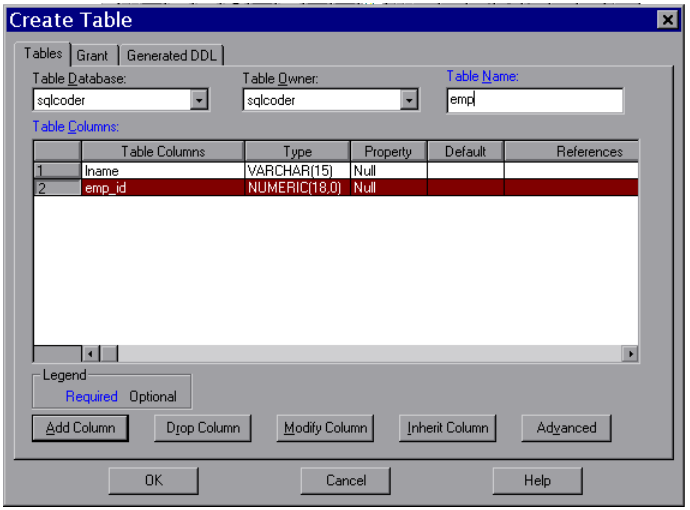
The Defining Column dialog displays again.

9 In the Name field, type **EMP_ID**.

10 In the Type field, choose **NUMERIC**.

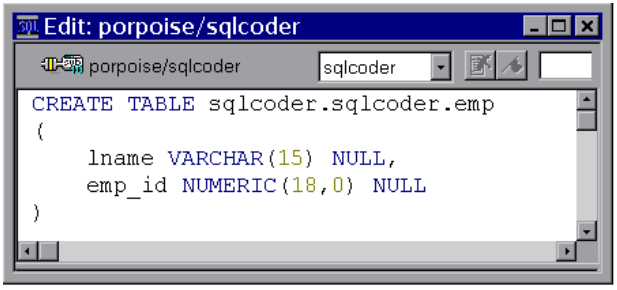
11 Click **OK** to add the emp_id field.

The Add Table wizard should now look as follows (with your Database and Owner names substituted):



12 Click the Generated DDL tab to view the DDL. Notice that the statements are based on the information you have entered in the previous steps.

13 Click **OK** to transfer the DDL code to a new Edit Window.



You can now execute, modify, or save the DDL to a file.

14 To execute the create table statement, verify the new Edit Window has focus and click the Execute toolbar button . Coder executes the statement and creates the **emp** table.

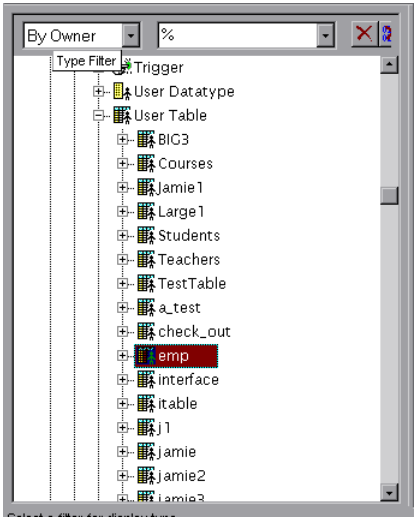
Inserting Data Through the Catalog Browser

There are multiple ways to insert data into a table. In this section, you use the Coder editable grid to insert data into the **emp** table.

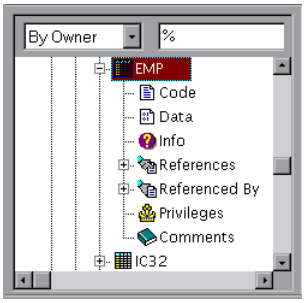
1 Switch to the Catalog Browser, and click the right mouse-button to display the menu.

2 Select **Refresh** to view the server in its current state.

3 Navigate through the Catalog Browser to find the **emp** table you created in the previous section: double-click the database, account name, and the User Tables icon.



4 Double-click **emp** to display the associated actions:



5 Double-click the Data icon to display a data grid in the Application Workspace.

The data grid appears with [NULL] values in the fields.

6 Highlight the first cell (in the lname field) and type **Lerner**.

Coder inserts a new row in the grid.

7 Highlight the cell under emp_id and type the value **001**.

8 Click <Return> to enter the value in the cell.

9 To specify another row to insert, click in the [NULL] field under lname.

10 Type **Heath** and **002** in the lname and emp_id fields, and press the <Return> key.

11 Select **File, Close Data** to close the Edit Window and commit the transaction.

12 Verify the data was inserted by doing one of the following:

- Click the Data icon under the **emp** table.
- Open a new Edit Window, type the statement **Select * from emp**, and click the Execute button .

4

The Catalog Browser

This chapter describes the Catalog Browser: the versatile navigation and execution facility included with Coder.

Overview **4-3**

Getting Started **4-4**

The Contents of the Catalog Browser **4-4**

Catalog Browser Icons 4-8

Navigating in the Catalog Browser **4-10**

Right-click Popup Menu 4-12

Applying Filters **4-13**

By Owner 4-14

By Type 4-14

By Name 4-15

Adding and Switching Connections **4-16**

Adding Connections 4-16

Switching Connections 4-16

Dropping Objects **4-17**

Generating Scripts and Procedures **4-17**

Generating a DML Statement for a Table 4-18
Generating a Procedure for a Table 4-18
Generating an Insert Script for a Table 4-19

Reverse-Engineering 4-22
From the Catalog Browser 4-22
Using the SQL Scripting tool 4-22

Viewing and Inserting Data 4-22

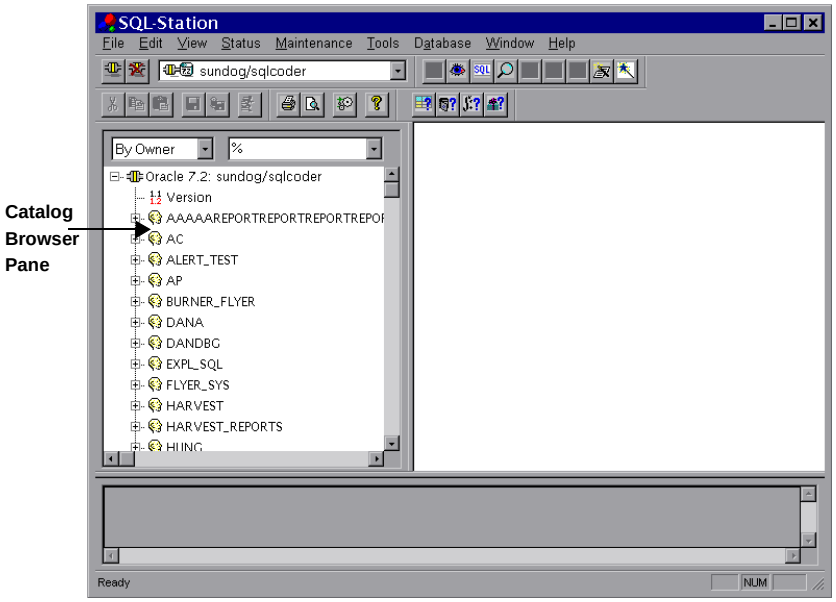
Executing Stored Procedures and Functions 4-23

Overview

The Catalog Browser is a versatile environment in which you can view the contents of a database server, examine the structure, dependencies, and code of the listed objects, and execute or otherwise interact with server objects.

You can use the Catalog Browser to do the following:

- Reverse-engineer objects
- Navigate server objects on multiple connections
- View objects through a variety of filters
- Execute stored procedures, functions, and packages
- Recompile server objects
- Generate an Insert Script to replicate the data in a table
- Generate template INSERT, UPDATE, SELECT, or DELETE Statements, Procedures, or complete Packages on a specific table
- Drop objects
- View dependencies
- View code in a sophisticated editor
- Invoke the SQL-Station debugger tool (if installed on your system)
- Delete data
- View and modify table data



Getting Started

After you have started Coder and established at least one connection, a Catalog Browser is automatically opened. To open a second Catalog Browser pane, select **Tools, Catalog Browser** from the Coder main window, or click the button.

The Catalog Browser can exist as a pane in the Coder main window (as in the above figure), or you can double-click the Catalog Browser border to make it a floating window.

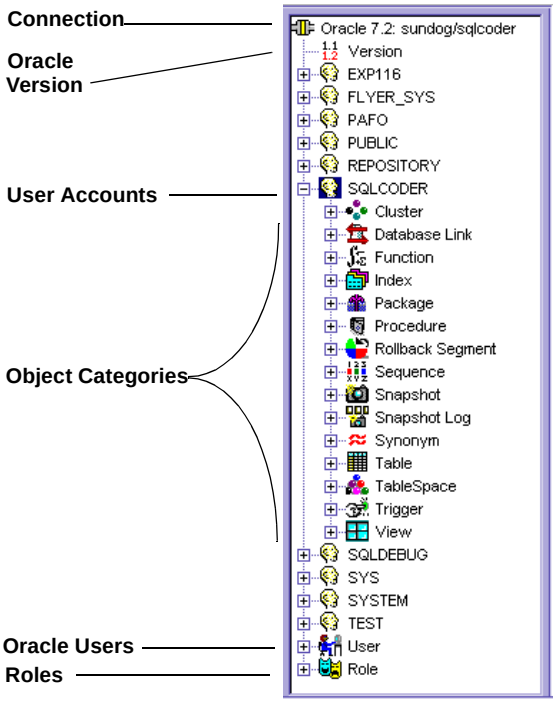
The Contents of the Catalog Browser

The Catalog Browser displays a connection icon for each active connection. You can expand a connection icon to display all User Accounts (Oracle) or databases (SQL Server) for that connection. On a SQL Server connection, expand a database to see its user accounts, and expand each user to see a list of object categories (such as User

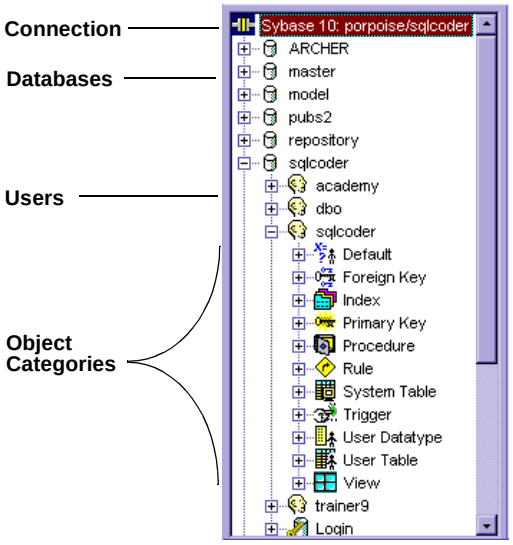
Table, Procedure, or View). On an Oracle connection, expand a User Account  to see a list of object categories (such as Table, Procedure, or View). You can then expand each object category to display the specific objects of that type on the selected User Account.

Note • **This assumes you are viewing the Catalog Browser in its default state of Filter By Owner. If you specify Filter By Type, the connection icons expand differently. For more information, see the *Applying Filters* section of this chapter.**

Note • **Your connection ID determines which objects are visible to you. You will only see objects for which you have adequate permissions.**



Catalog Browser Icons (Oracle Connection)



Catalog Browser Icons (SQL Server Connection)

Catalog Browser Icons

Oracle Object Types


Clusters


Database Links


Functions


Indexes


Packages


Procedures


Rollback Segments


Sequences


Snapshots


Snapshot Logs


Synonyms


Tables


Triggers


Views


Users


Roles


TableSpaces

SQL Server Object Types


Defaults


Foreign Keys


Indexes


Primary Keys


Procedures


Rules


System Tables


Triggers


User Datatypes


User Tables


Views


Login


User


Group

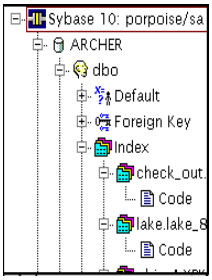

Alias


Threshold


User Message

Navigating in the Catalog Browser

The Catalog Browser shows an icon for each active server connection. To navigate through objects on a server, double-click the connection icon for that server. Depending on the type of icon currently selected, and the type of filter applied, each icon expands to display different nodes. For example, when expanding the Catalog Browser tree with the default By Owner filter, objects expand as follows:



- Connection icons expand to show owners/databases
- Database icons expand to show owners (SQL Server)
- Owner icons expand to show types of owned objects (such as Indexes, User Tables, Procedures, or Views)
- Object icons expand to show a list of specific objects of that type
- Specific object icons expand to show a list of components and relevant actions

Depending on the type of specific object you are expanding, a different list of component and relevant actions display. The following table shows common components:

Component	Icon	Action When Clicked
Code		Displays the SQL code for the selected item in a separate Edit window.
References		Displays the objects that the selected object references.
Referenced By		Displays the objects that reference the selected object.
Privileges		Shows which users or roles have privileges on the selected object.
Data		Displays the contents of a table or view in an editable grid in the Application Workspace.
Index		Displays the indexes on the selected table or view.
Trigger		Displays the triggers that reference the selected object.
Comments		Displays a dialog in which you can view or add comments about the selected object
Info		Displays column information for the selected table or view in grid format.

Right-click Popup Menu

Additionally, you can choose context-sensitive options from the right-click popup menu:

Right-click Popup Menu Option	Valid for Object Type(s)	Action
Drop	All	Drops the selected object from the schema
Execute	Function, Procedure, Package Procedure, Package Function	Executes the selected object, prompting for input parameters.
Recompile	Oracle Function, Procedure, or Trigger	Recompiles the object on the active connection.
Update Statistics	SQL Server Tables	Updates the System Catalogs.
Generate Insert Script	Tables	Generates a fully runnable INSERT statement based on the selected table and its associated data.
Generate Procedure	Tables	Generates a template for an INSERT, SELECT, UPDATE, or DELETE procedure based on the selected table.
Generate Statement	Tables, Oracle Triggers	Generates a template for an INSERT, SELECT, UPDATE, or DELETE statement based on the selected table (or for Triggers, a template for a SELECT statement).

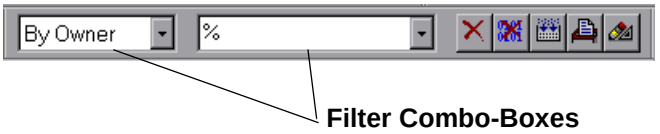
Right-click Popup Menu Option	Valid for Object Type(s)	Action
Generate Package	Oracle Tables	Generates a package body based on an Oracle Table.
Delete Data	Tables	Deletes all data from the selected table.
New	All	Invokes the SQL Wizard. You can use the SQL Wizard to create the DDL for a new object.
Debug	Oracle Procedures, Packages, Triggers, or Functions	This option appears when you have also installed SQL-Station Debugger. Use it to invoke the Debugger on the selected object.
Refresh		Use to refresh the Catalog Browser display.
Compare		Compares the reverse-engineered code for two simultaneously selected objects.

Applying Filters

You can apply filters within the Catalog Browser to change the way information is displayed. The following filters are available:

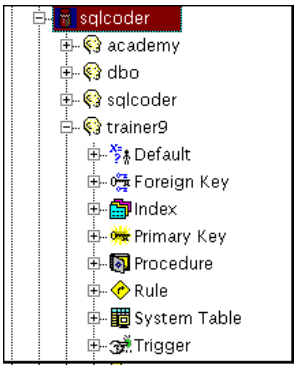
- By Owner (the default)
- By Type
- By Name

Use the filter combo-boxes on the Catalog Browser toolbar to change between filter types:



By Owner

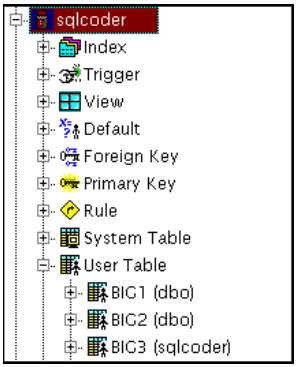
By default, the Catalog Browser displays objects By Owner. In this case, each user appears as an upper-level node in the tree, and beneath each user are the objects created/owned by that user.



Sorted By Owner

By Type

The By Type filter displays each object type as an upper-level node in the tree, and beneath each object type are all objects of that type, regardless of creator/owner:



Sorted By Type

When you sort objects By Type, the owners are displayed after the object name in parenthesis.

Note • **If the object is a function or procedure, the owner name is in the second set of parenthesis. The first set of parenthesis for those object types contain arguments.**

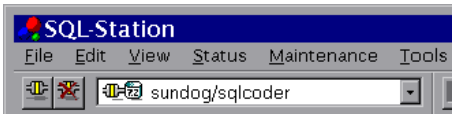
By Name

Use the Name Filter to *restrict* the objects displayed in the Catalog Browser. When you apply a name filter, only objects whose name meets the filter criteria are displayed. The default value in the Name Filter combo-box is %, the database wildcard symbol for *all objects*.

You can specify characters, wildcards, or a combination of both in the Name Filter combo-box.

Adding and Switching Connections

At any point in a Coder session, you can add connections or switch between existing connections. The *active connection* is the connection displayed in the list-box at the top of the Coder Main window.



Adding Connections

To add a new connection, follow these steps:

1 Click the connect toolbar button or select **Database, Open Connection** from the menu. The Login dialog appears.

2 Enter the connection information and click **Connect**.

The connection is established, and the list-box at the top of the screen displays the new connection as the active connection.

Note • **The active connection is inherited by a new Edit Window at the time the Edit Window is opened. For more information, see *The Edit Window* chapter.**

Switching Connections

When you have established more than one connection, you can use the list-box at the top of the Coder Main window to switch between them. Simply select the desired connection to make it active.

Note • **The active connection is inherited by an Edit Window at the time it is opened. For more information, see *The Edit Window* chapter.**

Note • **You can always view multiple connections in the Catalog Browser, however to change the *active* connection, you must switch to it in the Active Connection toolbar. Simply expanding that connection in the Catalog Browser will not change the active connection.**

Dropping Objects

You can drop an object from within the Catalog Browser by doing one of the following when the object is highlighted:

- Right-click and select **drop** from the popup menu
- Press the **Del** key
- Click the Drop An Object toolbar button  from the Catalog Browser toolbar (you might need to resize the Catalog Browser to see this button)

You are prompted to confirm the operation, and the object is dropped from the server.

Note • **You can select multiple object simultaneously by holding down the Control key while you select them.**

Generating Scripts and Procedures

You can generate insert statements, DML statements, and procedures based on tables from within the Catalog Browser. You can generate statements and procedures of the following types:

- INSERT
- UPDATE
- SELECT
- DELETE

Generating a DML Statement for a Table

Follow these steps to create a DML statement for an existing table:

1 In the Catalog Browser, expand a connection (that contains tables) to the table category level.

2 Highlight a tablename.

Note • **Select a table that you own, or on which you have adequate permissions.**

3 Click the right mouse-button, and select **Generate Statement**. Select the statement type to generate (**Select**, **Insert**, **Update**, or **Delete)**.

Note • **If you have selected a table that doesn't have a unique index or primary key, only Select is enabled. This is to prevent accidentally alteration the selected table.**

4 Coder opens a new Edit Window and generates the corresponding DML statement into it.

5 Switch to the Edit Window to modify the statement. You can execute the statement to the associated connection or save it to a .sql file. (For more information on these actions, see *The Edit Window* chapter.)

Generating a Procedure for a Table

Follow these steps to create a procedure for an existing table:

1 In the Catalog Browser, expand a connection (that contains tables) to the table category level.

2 Highlight a tablename.

Note • **Select a table that you own, or on which you have adequate permissions.**

3 Click the right mouse-button, and select **Generate Procedure**. Select the procedure type to generate (**Select**, **Insert**, **Update**, or **Delete)**.

4 Coder opens a new Edit Window and generates the corresponding Create Procedure statement into it.

5 Switch to the Edit Window to modify the statement. You can save it to a .sql file, or execute the statement to compile it on the associated connection. (For more information, see *The Edit Window* chapter.)

Generating an Insert Script for a Table

From within the Catalog Browser, you can generate an Insert script for a selected table. Coder generates a script to duplicate the table structure and the data contained in the table.

Follow these steps to generate an INSERT script for a table:

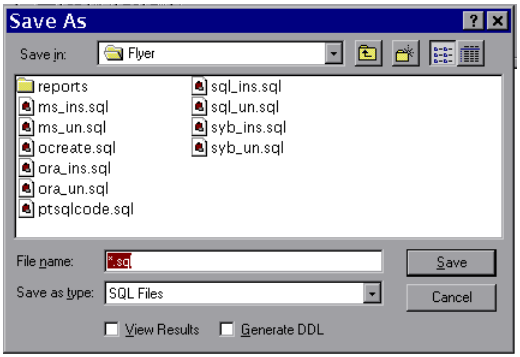
1 In the Catalog Browser, expand a connection (that contains a table) to the table category level.

2 Highlight a table name.

Note • **Select a table that you own, or on which you have adequate permissions.**

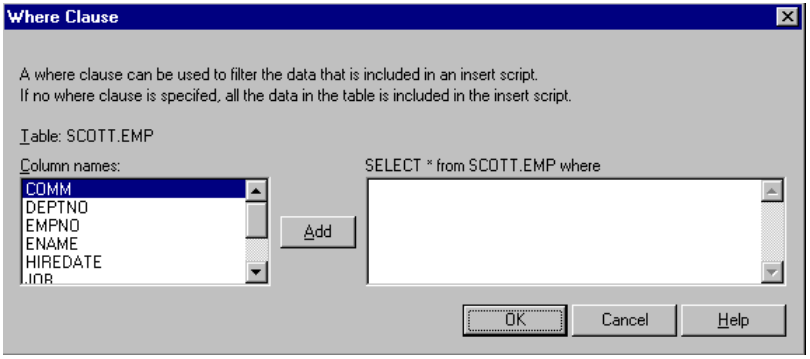
3 Click the right mouse-button, and select **Generate Insert Script**.

The Save As dialog appears:



4 Specify a filename for the insert script. Check the **View Results** box to display the saved insert script in an Edit Window, and check **Generate DDL** if you want the insert script to include the CREATE statement for the table.

5 Click **Save**. The Where Clause dialog appears:

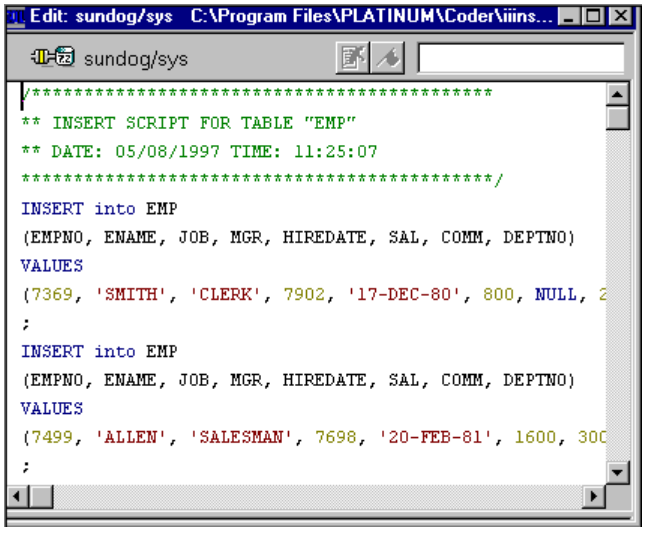


Use this dialog to filter the data that is included in the script. If no Where clause is specified, all data will be included.

6 If you want to specify a Where clause:

- Select one or more column names in the Column Names field and click **Add** to move them to the right pane.
- Edit the right pane to specify the condition to place on the column(s), such as **empno > 008**.

7 Click **OK**. The insert script is saved to the specified file. If you have checked **View Results**, the file is also displayed in a new Edit Window.



If you have not chosen to View Results, you can open the saved .sql file in an Edit Window.

- Select **File, Open** from the menu and choose the filename you specified in step 4, above.

Reverse-Engineering

Coder can reverse-engineer any object from within the Catalog Browser. *Reverse-engineering* allows you to view the source code used to create an object. You can reverse-engineer an object to see how it was created, or to create an identical or similar object without having to type in all of the DDL statements.

From the Catalog Browser

To reverse-engineer an object from the Catalog Browser, follow these steps:

1 Locate the object in the Catalog Browser tree.

2 Expand the icon for the object to display the **Code** icon .

3 Double-click the **Code** icon. Coder reverse-engineers the object into a new Edit Window (based on the original connection, not the active connection). You can switch to the Edit Window to view, edit, run, or save the code.

Using the SQL Scripting tool

To reverse-engineer an object or multiple objects simultaneously, use the SQL Scripting utility. This utility is available from the **Tools** menu or the Scripting Tool toolbar button .

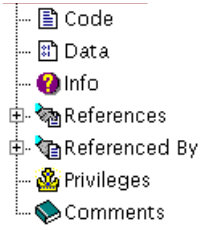
For instructions on using the SQL Scripting utility, see the *SQL-Station Coder Tools* chapter.

Viewing and Inserting Data

From the Catalog Browser, you can use the Data icon to view existing data in a table, or to add or modify data.

1 Navigate to the table level in the Catalog Browser.

2 Select a tablename and double-click to display the associated action icons:



3 Click the Data icon to display a Data Grid in the Application Workspace.

If the table contains data, you can view or modify it.

- To modify the data, type in any field.
- To add data, type values in an empty row.

Note • **Under Oracle, the rows are not inserted until you close the Edit Window, or you click the Commit button .**

Executing Stored Procedures and Functions

Coder provides a mechanism for executing stored procedures and functions from within the Catalog Browser. You can access the procedure execution facility from the **Tools** menu, the Catalog Browser right-mouse popup menu, or the Procedure execution toolbar button . See the *Procedure Execution Utility* chapter for more information.

5

The Edit Window

This chapter describes the Coder Edit Window. You can use the Edit Window to open, edit, save, and run SQL commands.

Overview **5-2**

Opening an Edit Window **5-3**

Entering and Running SQL Statements **5-4**

Setting Edit Window Preferences **5-6**

Object Lookup Utility **5-6**

Codewright© Features **5-7**

Editor Emulation 5-7

Find 5-9

Quick Search 5-9

Version Control 5-10

Syntax Highlighting 5-11

Brace Matching 5-11

Spell Checking 5-12

API Assistance 5-12

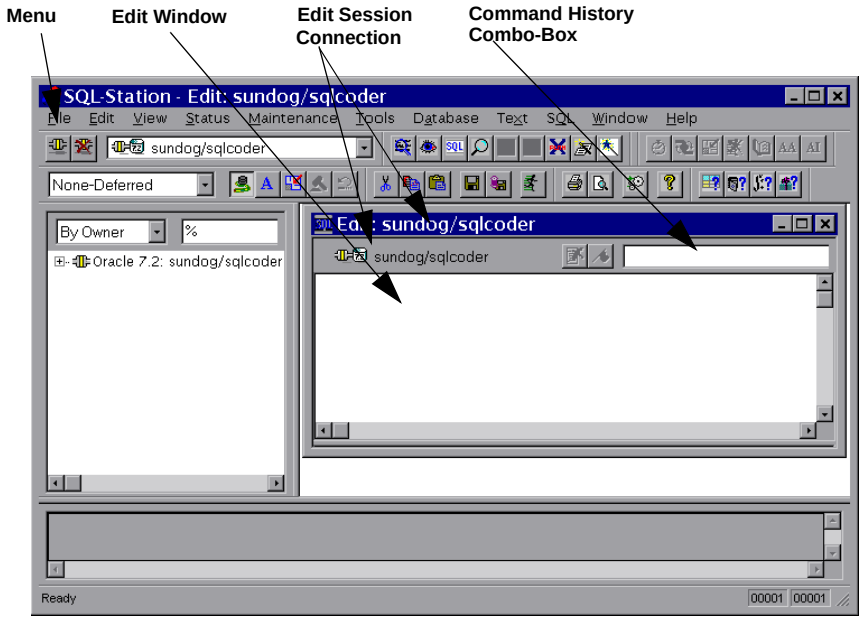
Code Templates **5-13**

Overview

Coder includes a SQL Editor in which you can edit and run SQL statements, as well as saving them to or retrieving them from files. The Coder editor is based off of the Codewright© editor with many sophisticated features, including:

- Syntax Highlighting
- VI, CUA, EMACS, and BRIEF emulation capability
- Grep-like Find utility
- Integration with Source Code Control Systems
- Code templates
- Comment templates
- Go to line
- Find/Replace
- Brace Matching
- Spell Checking
- Drag and Drop

The Coder Edit Window appears as follows:



Opening an Edit Window

After you have started Coder, you can open an Edit Window regardless of whether you have established a connection to a database server. If you haven't yet connected to a database server, however, you will only be able to save, edit, or create SQL statements; you cannot run code against a server until you connect.

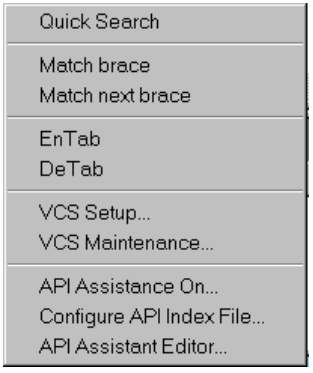
To open an empty Edit Window, do one of the following:

- Select **File, New**
- Select **Tools, Edit Window**
- Click the Edit Window toolbar button 

Note • You can also open an Edit Window from the Catalog Browser by clicking the **code** icon associated with an object. For more information, see the *Reverse-Engineering* section of *The Catalog Browser* chapter.

Coder displays a new Edit Window.

When the cursor is in the Edit Window, you can click the right-mouse button to view a popup menu:



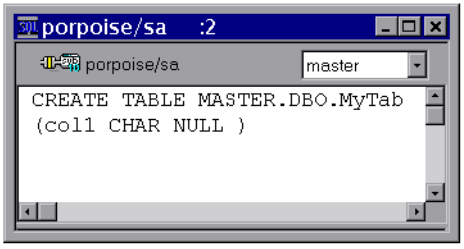
Entering and Running SQL Statements

You can enter SQL statements into a new or open Edit Window in these ways:

- Select **File, Open** from the menu to open an existing .sql file.
- Type directly into the active Edit Window.

- Use a SQL Wizard to construct SQL statements. When you select **OK**, the code is inserted into a new or existing Edit Window (for more information, see the *SQL Wizards* section of the *SQL-Station Coder Tools* chapter).
- Use the SQL Scripting utility to reverse-engineer a database object or objects. The code for the selected object is displayed in a new Edit Window. (For more information, see the *Generate SQL Script* section of the *SQL-Station Coder Tools* chapter.)
- From within the Catalog Browser, click the **code** icon for an object in an expanded browser tree. The code for the associated object is displayed in a new Edit Window. (For more information, see *The Catalog Browser* chapter.)

Unless opened from the Catalog Browser, an Edit Window inherits the connection that is active at the time it is opened. For example, if the active connection is to **porpoise/sa**, any Edit Window that is opened will be associated with the **porpoise/sa** connection:



When you open a new Edit Window from the Catalog Browser, it is associated with the connection for the selected object.

Regardless of how it is opened, under SQL Server you can dynamically specify the database to which to execute by using the database listbox at the top of the Edit Window.

Once an Edit Window contains SQL statements, you can execute them against the associated server. To run the entire contents of the Edit Window, select **SQL, Execute** from the menu, or click the Execute  button. To run a selection from the Edit Window, highlight the selection before executing.

Setting Edit Window Preferences

You can use the Edit Window tab and General tab on the Preferences dialog to specify editor and display options for the Edit Window. For detailed information on setting Edit Window preferences, see the chapter *Setting Coder Preferences*.

Object Lookup Utility

The Coder object lookup utility lets you look up information about tables, procedures, functions (Oracle), and packages (Oracle) for the active connection. Use the lookup utility to do the following:

- Drag and drop object names into an edit window
- Generate SQL statements on selected objects
- Look up information about an object, such as (depending on the object type):
  - name
  - comprising columns
  - datatypes
  - length
  - precision
  - scale
  - default
  - owner

- database (SQL Server)

For more information about the Object Lookup utility, see the *SQL-Station Coder Tools* chapter.

Codewright© Features

The Coder Edit Window incorporates features of the Premia Codewright editor, such as:

- Editor emulation (VI, EMACS, Brief, CUA)
- Find and Quick Search
- Version Control
- Syntax Highlighting
- Brace Matching
- API Assistance

Coder includes Codewright's online help. You can access it from the **Help** menu for detailed information about Codewright features and functionality. This section outlines some of the Codewright features.

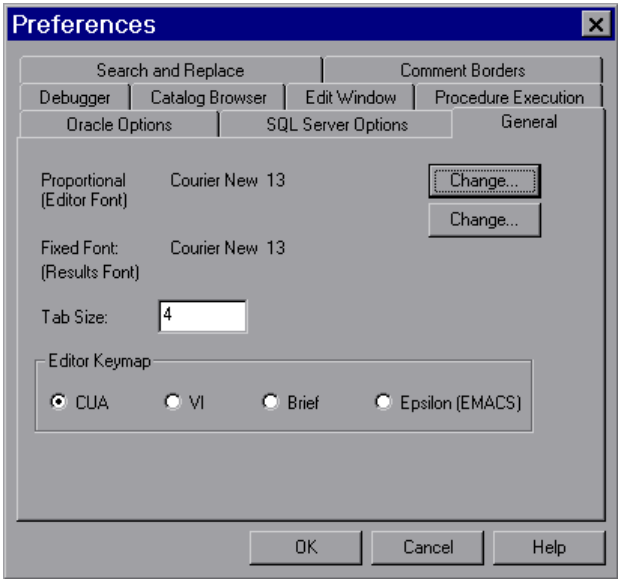
Editor Emulation

The Codewright editor is capable of emulating the following editors:

- BRIEF
- Epsilon (EMACS)
- vi
- CUA

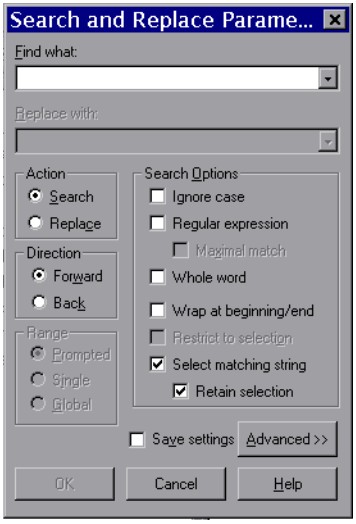
By default, the Codewright editor is Common User Access (CUA) compliant. This command set offers a number of short-cut keystrokes to allow you to bypass the menus.

To switch between editor modes, use the General tab of the Preferences dialog (**Edit, Preferences**, or click the Preferences toolbar button):



Find

You can search for a text string in an Edit Window. Use the **Find** option on the **Edit** menu to search through a file for a string occurrence.



You can also use the Preferences dialog to specify default values for the Search and Replace dialog. For more information, see the chapter *Setting Coder Preferences*.

Quick Search

You can use the Quick Search utility to perform a limited version of the Find option. Quick Search locates the next occurrence of the word at the cursor location. Quick Search is case-sensitive.

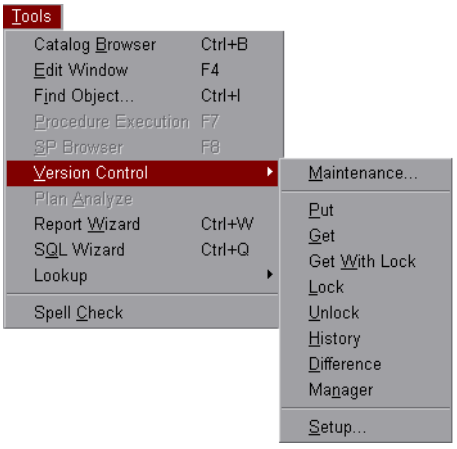
To use Quick Search, click the right mouse-button in an Edit Window, and choose **Quick Search** from the popup menu.

Version Control

The Coder **Tools** menu contains a **Version Control** option. This allows Coder to work with your version control software in two ways:

- Using a command line interface
- Using your provider's Application Programming Interface (API)

To select which to use, choose **Tools, Version Control, Setup** from the main menu:



You can also use the Edit Window right-click popup menu and select **VCS Setup**.

Command Line Interface

To use the command line interface for Version Control, you need a version control system that processes arguments on the command line to tell it what file to operate on, and what options you are specifying.

API

To use your provider's API, you need the following:

- A version control system that offers a robust API
- An interface DLL, supplied by the provider

Select **Help, Codewright Help** from the main menu for specific information about configuring Version Control systems.

Syntax Highlighting

The Edit Window automatically displays statement-sensitive color coding for SQL statements. Database keywords, such as SELECT, INSERT, and so on, are colored differently for easy identification.

Extending Highlighting to Other File Types

You can configure Coder to highlight keywords in files with an extension other than **.sql**. To add file types you must edit the cwright.ini file, installed with Coder:

1. Using a text editor (such as notepad), open the **cwright.ini** file, located in the directory in which you installed Coder.
2. Scroll down to the [Extension] section (near the end of the file).
3. Add the following line to this section:

   `ExtAlias=.`*`xxx`*`,.sql`

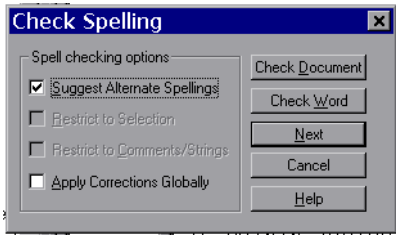
   where **xxx** is the file extension you want to include.

Brace Matching

Brace Matching verifies that there is a closing brace for every opening brace. Access the Brace Matching feature from the Edit Window right-click popup menu.

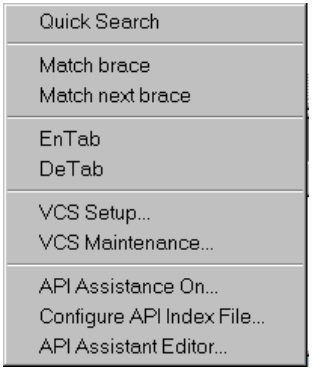
Spell Checking

Use the **Spell Checking** option from the **Tools** menu to check the spelling in the active Edit Window.



API Assistance

This feature helps you code a call to a subroutine or function. Access API Assistance from the **Help** menu or from the Edit Window right-click popup menu.



Code Templates

Coder provides a selection of PL/SQL and Transact-SQL code templates. With a single button-click, the structure for the object is generated and copied to an Edit Window. You can then edit the code and execute it to create the object.

Table 5-1 • Code Templates

Template	Oracle	SQL Server
Create Function Shell	●	
Create Index Syntax		●
Create Package Shell	●	
Create Procedure Shell	●	●
Create Procedure Syntax		●
Create Table Shell	●	●
Create Table Syntax		●
Create Trigger Shell	●	
Create Trigger Syntax (complex)		●
Create Trigger Syntax (simple)		●
Cursor Loop (complex)	●	
Cursor Loop (simple)	●	
Exception Handler (simple)	●	
For Loop	●	
Function Comment	●	
Goto (complex)	●	
Goto (simple)	●	
Goto syntax		●
If (simple)	●	●

Table 5-1 • Code Templates

Template	Oracle	SQL Server
If (complex)	●	●
Local Variable and Assignment Syntax		●
Loop	●	
New Datatype		●
New Rule		●
Package Body Comment	●	
Package Comment	●	
Print Syntax		●
Procedure Comment	●	●
Raiserror Syntax		●
Standard Exception Handler	●	
Table Comment	●	●
Table Type	●	
Trigger Comment	●	●
Waitfor Syntax		●
While Loop	●	
While Syntax (simple)		●
While Syntax (complex)		●

Opening Templates

You can invoke the Available Code Templates dialog by pressing **Control-t** from an open Edit Window, or by selecting **Text, Insert Template** from the menu.

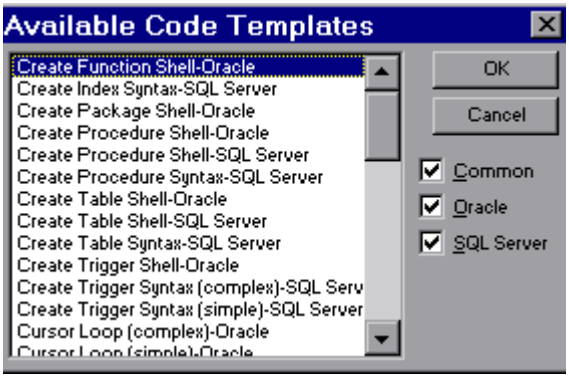


Figure 5-1 • Available Code Templates dialog

Double-click a template name (or highlight it and click OK) to copy the template to the active Edit Window. There, you can revise it and execute it against a database server, or save it to a .sql file.

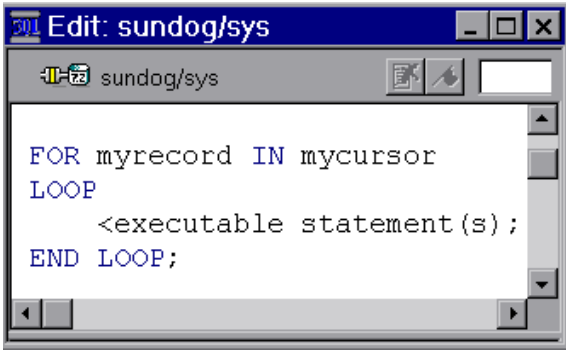


Figure 5-2 • Generated For Loop template

Editing Templates

Use the Edit Templates dialog to add, delete, or customize a code template.

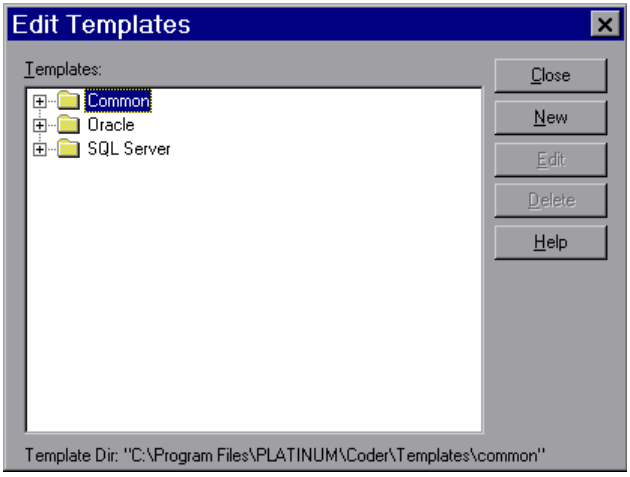


Figure 5-3 • Edit Templates dialog

When you add a template, it is saved in the Templates subdirectory of the Coder installation directory (with a **.tpl** file extension).

When you delete a template, a copy of the deleted template is saved in a Backup subdirectory of the Templates directory.

The CodeWright online help that is provided with Coder contains information about CodeWright's template macro language. To access CodeWright's online help, select **Help, CodeWright Help** from the Coder menu.

6

SQL-Station Coder Tools

This chapter describes the Generate SQL Script, Find, Compare, SQL Wizard, and Object Lookup tools available with Coder.

Overview **6-2**

Find Object **6-3**

Compare **6-5**

Generate SQL Script **6-7**

SQL Wizards **6-10**

Accessing the SQL Wizards 6-11

Object Lookup Utility **6-12**

Table Lookup 6-13

Function Lookup 6-14

Package Lookup 6-15

Procedure Lookup 6-16

Overview

The Coder **Tools** menu contains the following options:

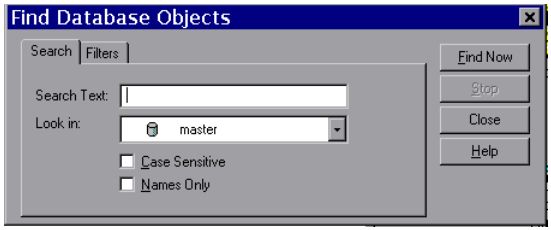
- Compare
- Catalog Browser
- Generate SQL Script
- Edit Window
- Find Object
- Procedure Execution
- SP Browser
- Version Control
- API Assistant Editor
- Plan Analyzer
- Debug
- Report Wizard
- SQL Wizard
- Lookup
- Spell Check

This chapter describes the options **Generate SQL Script**, **Find Object**, **SQL Wizards**, **Compare**, and **Lookup**. The remaining options are discussed in other chapters.

Find Object

Use the Find Object utility to locate a database object on the active connection. Follow these steps to use the Find Object utility:

1 Select **Tools, Find Object** or click the Find Database Objects toolbar button . The Find Database Objects dialog displays:

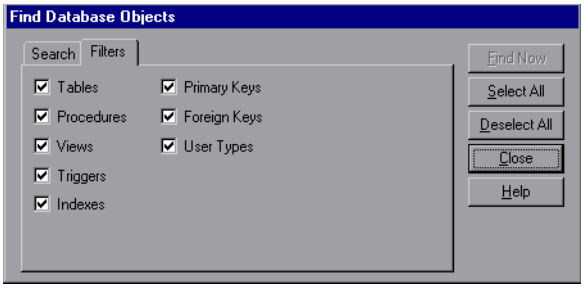


The Find Database Objects dialog contains two tabs: **Search** and **Filter**. Use the fields on these tabs to narrow down the search.

Note • **The Find Database Objects dialog inherits the active connection. If you want to find an object on a server other than that in the active connection listbox, change the active connection before invoking the Find Database Objects dialog. For more information about switching the active connection, see *Switching Connections* in the *The Catalog Browser* chapter.**

2 Use the fields on the Search tab to specify any Search Text, the database or schema to look in, and whether the Search Text should be interpreted literally by case or name. You can use standard database wildcards (such as %) in the Search Text field.

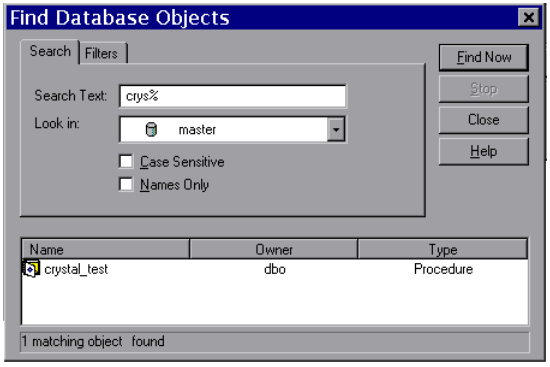
3 Click the Filters tab to display the available object type filters:



4 By default all object types are checked. Click the checkboxes to toggle between selected and deselected for any object type. Click **Select All** or **Deselect All** to select or deselect all objects automatically.

5 After you have specified the search parameters on both tabs, click **Find Now** to start the search.

When you initiate the search, Coder displays any objects found on the active connection that match the criteria you have specified:



The object owner is displayed to facilitate searching for the object in the Catalog Browser.

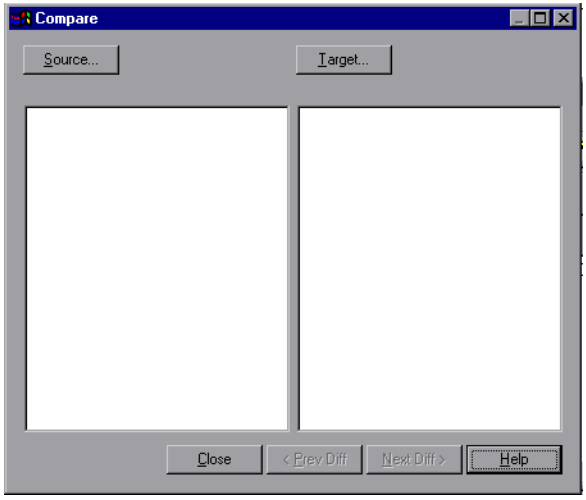
Compare

Use the Compare option to compare two database objects, disk files, or a database object and a disk file in an active connection.

To use Compare:

1 Select **Tools, Compare** from the Coder Main menu.

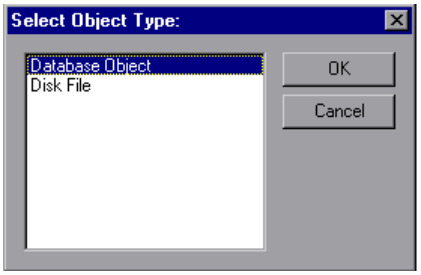
The Compare dialog appears:



The Compare dialog contains two buttons: **Source** and **Target**.

2 Click **Source**.

The Select Object Type dialog appears:



3 Select **Database Object** or **Disk File** and click **OK**.

If you selected **Database Object**, the Select Database Object dialog displays a Catalog Browser to choose from.

- Select a database object from the Catalog Browser and click **OK**.

OR

If you selected **Disk File**, the Open Source File dialog appears.

- Select a source file and click **OK**.

The contents of the file display in the Source pane.

4 Click **Target**. Repeat step 3 to select and insert an object type (database object or disk file) in the Target pane to compare with the object type displayed in the Source pane.

The differences between the data in the two panes are highlighted in red.

Note • **You can also access the Compare utility from the Catalog Browser right-click popup menu when two objects are selected.**

Generate SQL Script

Use the Generate SQL Script option to reverse-engineer one or more database objects. You can then edit, save, or execute the generated code.

Follow these steps to use the Generate SQL Script option:

1 Select **Tools, Generate SQL Script** from the menu, or click the Scripting Tool button on the toolbar .

The Generate SQL Script dialog appears:

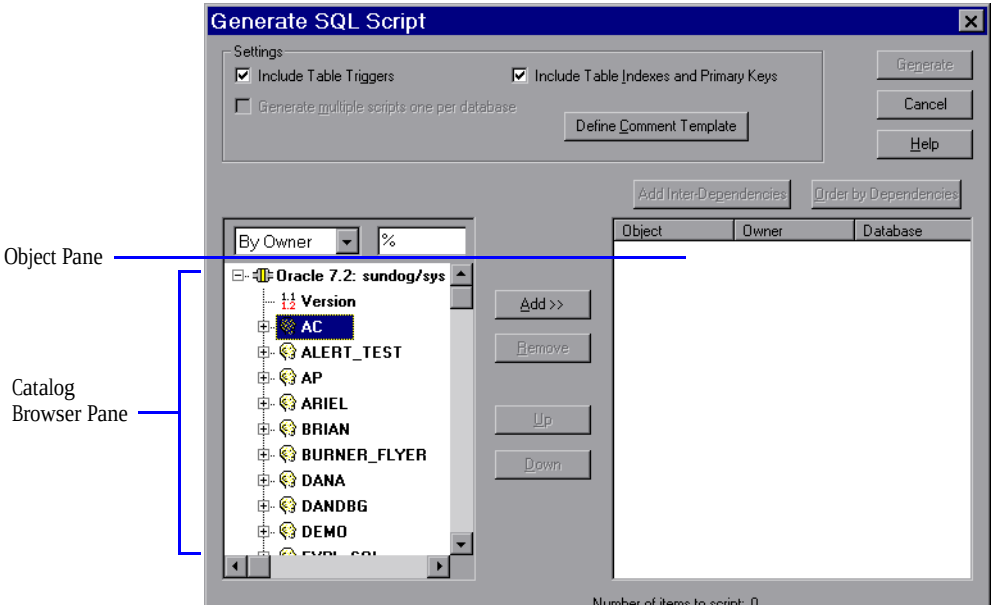
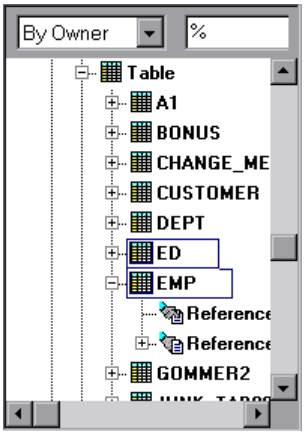


Figure 6-1 • Generate SQL Script dialog

The left pane of the dialog displays a copy of the Catalog Browser.

2 Navigate through the Catalog Browser pane until the object you want to reverse-engineer is displayed:



3 Highlight the object (or objects), and click **Add** to move it to the Object pane on the right side of the dialog.

Note • **You can hold down the Control key while clicking to select multiple objects.**

WARNING • **You cannot select objects from multiple connections or databases.**

4 Check the **Include Table Triggers** checkbox to include triggers in the generated SQL.

5 Check the **Include Table Indexes and Primary Keys** checkbox to include Indexes and Primary Keys in the generated SQL.

6 To include dependent objects in the generated SQL, click **Add Inter-Dependencies**. If there are dependent objects, they appear beneath the object name in the Object pane.

7 Click **Order by Dependencies** to reorder the objects in the correct logical execution order (for example, a table is defined before its indexes).

8 Click **Define Comment template** to invoke the Comments dialog. Here you can specify comments to appear at the top and prior to each object in the generated script.

When you are finished defining comments, click **OK**

9 Review the Object pane to verify it contains all the objects you want to include in your script. You can use the **Up** and **Down** buttons to change the objects' order in the pane.

10 Click **Generate**. Coder opens a new Edit Window containing the script. You can then edit, save, or execute the script.

Note • **The script is written to an Edit Window that is associated with the *original* connection (the connection from which the object was selected). This connection is displayed in the header of the Edit Window. If you choose to execute the script, it will be executed against the associated connection. To run the script against a different connection, save it and re-open it in an Edit Window associated with the desired connection.**

SQL Wizards

Coder's SQL Wizards facilitate the process of writing DDL for database objects. Simply enter information in the Wizard fields, and Coder generates the DDL statements necessary to create the object. You can then save, edit, or execute the statements.

Coder provides a different selection of SQL Wizards depending on the type of database connection.

Table 6-1 • Available SQL Wizards

SQL Wizard	Oracle	SQL Server
Anonymous Block	●	
Cluster	●	
Database Link	●	
Function	●	
Index	●	
Package	●	
Procedure	●	●
Role	●	
Rollback Segment	●	
Sequence	●	
Synonym	●	
Table	●	●
Tablespace	●	
Trigger	●	●
User	●	
View	●	

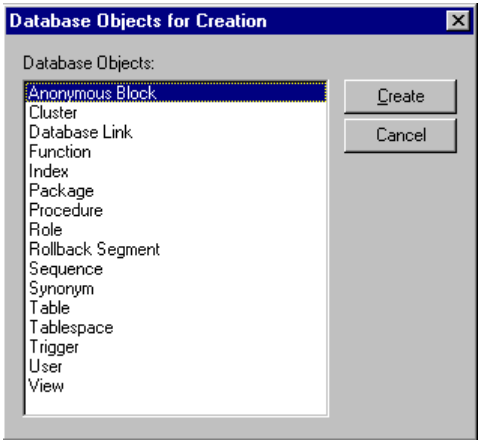
Accessing the SQL Wizards

You can access the SQL Wizards from the toolbar, the Coder **Tools** menu, or the Catalog Browser.

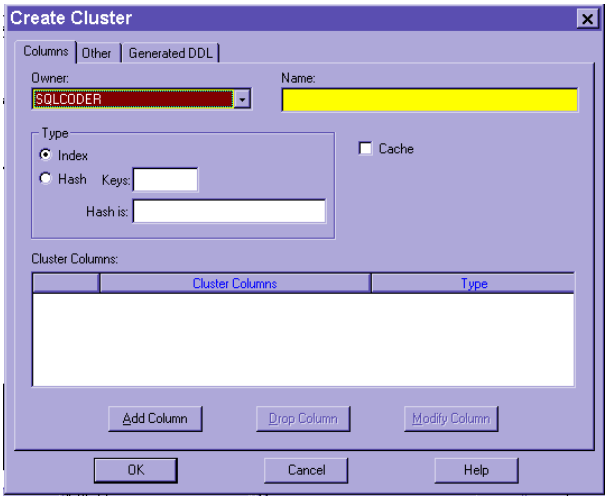
1 To invoke a SQL Wizard, do one of the following:

- Click the SQL Wizards toolbar button 
- Select **Tools, SQL Wizard** from the menu
- Select **New** from the Catalog Browser right-mouse popup menu when an object or object category is highlighted

A dialog displays the object wizards available for the active connection (or active Edit Window if one is open):



2 Select an object type and click **Create** to invoke a wizard.



3 Every wizard contains multiple tabs. Each tab lets you specify information about the object you want to create. A Generated DDL tab displays the DDL code as you enter information.

4 When you are done specifying information for the particular CREATE statement, click **OK** to move the generated DDL to a new or existing Edit window where you can then modify, save, and/or run it against a server connection.

Object Lookup Utility

The Object lookup utility lets you look up attributes for tables, functions, packages, and procedures. This feature is most useful from an Edit Window where you need information about specific object attributes, such as name or datatype.

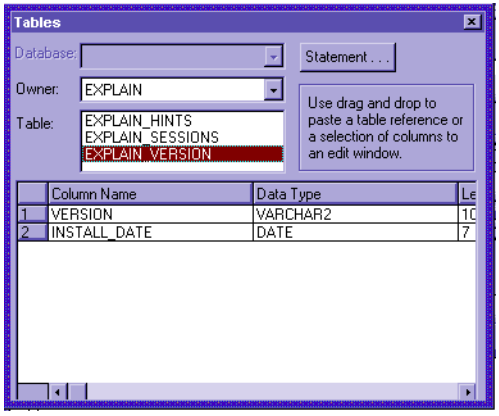
The Object lookup utility lets you

- look up tables, functions, packages, and stored procedures
- drag and drop object names into the Edit Window
- for tables, generate INSERT, SELECT, UPDATE, and DELETE statements

Table Lookup

Follow these steps to use the Table Lookup feature:

1 Select **Tools, Lookup, Table** from the menu or click the Table Lookup button from the toolbar. The Table Lookup dialog appears:



2 Scroll through the **Owner** combo-box to select the User Account containing the table you want to see. When you have selected a User Account, the table name(s) appear in the **Table** field.

3 Select the table from the **Table** field, and the contents of the table are displayed in the spreadsheet, including the following attributes:

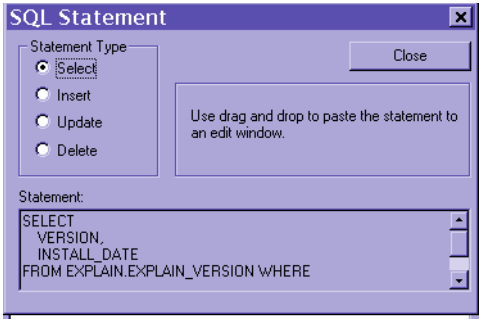
- Column Name
- Data Type
- Length
- Precision
- Scale
- Default
- Nullable
- Primary Key

Dragging and Dropping into an Edit Window

You can drag and drop a table name or one or more column names into an Edit Window. Highlight the table/column names you want to place in the Edit Window, and drag them until the cursor becomes a cross-hair. If you drag from the column names without highlighting an individual row, all column names are dragged when you move the cursor.

Generating a DML Statement

Click the **Statement** button to generate a DML statement based on the selected table. The SQL Statement dialog appears:

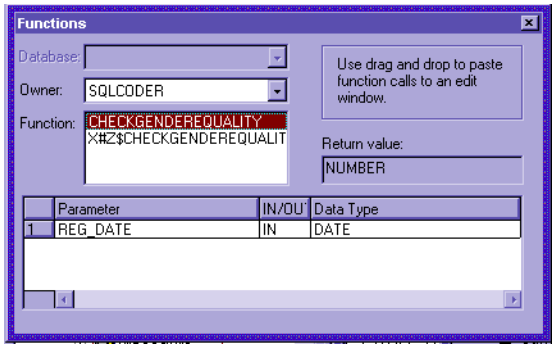


Drag and drop the statement into an Edit Window.

Function Lookup

Follow these steps to use the Function Lookup feature:

1 Select **Tools, Lookup, Function** from the menu, or click the Function Lookup button from the toolbar. The Function Lookup dialog appears:

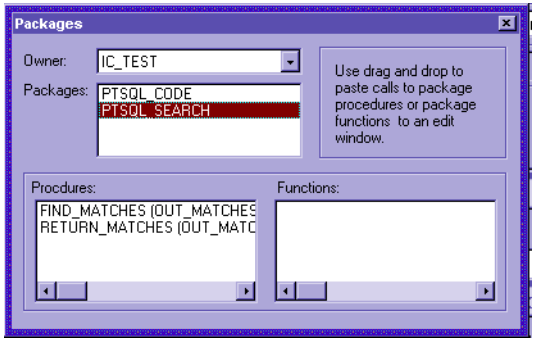


Drag and drop function calls into the Edit window.

Package Lookup

Follow these steps to use the Package Lookup feature:

1 Select **Tools, Lookup, Package** from the menu or click the Package Lookup button from the toolbar. The Package Lookup dialog appears:

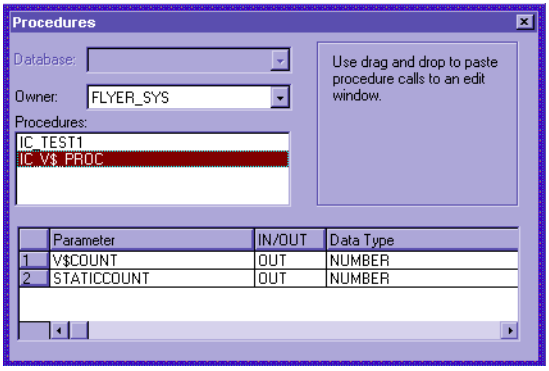


Drag and drop calls to package procedures or functions in the Edit Window.

Procedure Lookup

Follow these steps to use the Stored Procedure Lookup feature:

1 Select **Tools, Lookup, Procedure** from the menu or click the Procedure Lookup button from the toolbar. The Procedure Lookup dialog appears:



Drag and drop procedure calls to the Edit window.

7

Procedure Execution Utility

Coder provides a mechanism for executing Stored Procedures or Functions.

Overview **7-2**

Object Execution Utility **7-2**

Procedure Execution Preferences **7-5**

SQL Server System SP Execution **7-5**

Overview

Coder provides a method for executing stored procedures and functions. This enables you to run these objects directly from the Catalog Browser without a calling client-side application. The Coder input mechanism allows you to

- pass arguments to a procedure or function
- execute a procedure or function
- examine output parameters and return values
- view DBMS_OUTPUT (Oracle) and PRINT messages
- see server messages

Note • **Coder also provides a mechanism for compiling procedures and functions. This is a separate functionality from procedure and function** ***execution*****.**

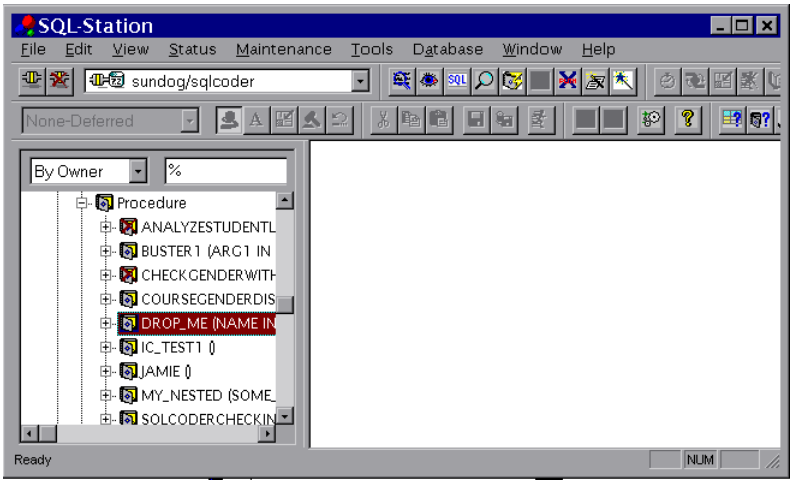
Coder also provides a utility for executing the SQL Server system stored procedures. For information about that utility, see the *SP Browser* chapter.

Object Execution Utility

To execute a stored procedure or function, the object must have been successfully compiled, and must be selected in the Catalog Browser. Access the Execution facility from the **Tools** menu, the Catalog Browser right-mouse popup menu, or the Procedure execution toolbar button  .

Follow these steps to execute a Stored Procedure or Function from the Catalog Browser:

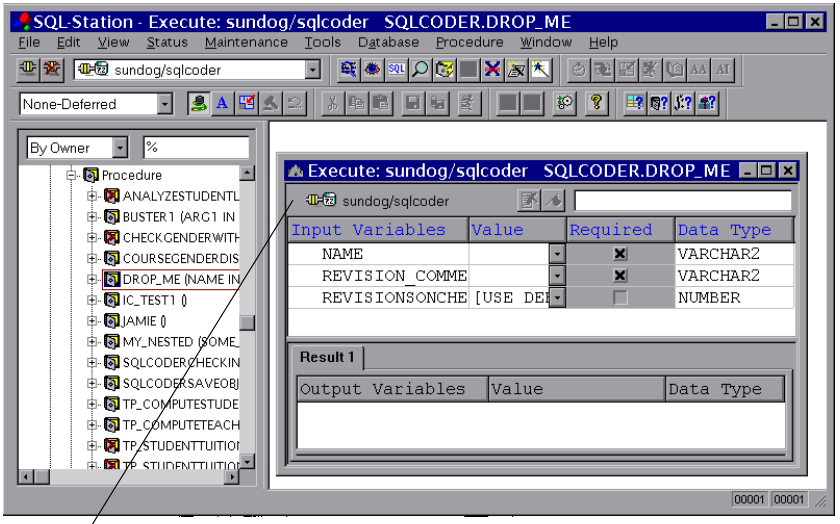
1 Expand the Catalog Browser to the level where the Procedure or Function is displayed:



2 Highlight the object and do one of the following:

- Click the right mouse-button and select **Execute**
- Select T**ools, Procedure Execution** from the main menu
- Click the Procedure Execution button on the toolbar

The Procedure Execution window displays in the Application Workspace:



Procedure Execution Window

The top of the Procedure Execution window displays the associated connection and input variable information, including variable values, datatypes, and whether they are required.

The bottom of the Procedure Execution window displays a Result tab that will show the values of output variables after the procedure or function is executed.

3 If the procedure or function takes input variables, specify the values in the top of the Procedure Execution window.

If the procedure doesn't take input variables, the procedure/function executes immediately.

4 Execute the procedure or function by doing one of the following:

- Click the Execute button
- Select **Tools, Procedure Execution** from the menu

The procedure executes. The values for any output variables or DBMS_OUTPUT or PRINT messages are displayed in the Result tab. Any Server messages are displayed in the server messages window.

You can use the Procedure Execution facility to run procedures or functions with different variable values and compare the results.

Procedure Execution Preferences

You can specify procedure-execution preferences. Use the Procedure Execution tab of the Preferences dialog to specify color-coding for Input and Output variables, and to specify the default Web browser for web execution. For more information, see the *Procedure Execution Tab* section of the *Setting Coder Preferences* chapter.

SQL Server System SP Execution

You can use Coder's SP Browser facility to easily run SQL Server system stored procedures. The SP Browser utility displays SQL Server system stored procedures in easy-to-understand formats such as **User Administration** and **Database Administration**. See the *SP Browser* chapter for more information.

8

Report Wizard

You can use the Coder Report Wizard to create a variety of object Summary, Code, and Dependency reports.

Overview **8-2**

Using the Report Wizard **8-4**

Overview

You can use the report wizard to generate information on Oracle server objects in a variety of configurations and formats. Invoke the report wizard from the Coder **Tools** menu.

Note • **For this release of SQL-Station, SQL Server reports are not supported.**

Table 8-1 • Available Reports

Report	Description
Table Summary	Lists information on tablespace and storage properties for selected tables.
View Summary	Lists information, including code, about selected views.
Cluster Summary	Lists all physical information about clusters. Results are grouped by Cluster Owner and ordered by Cluster Name.
Index Summary	Lists all physical information about index(es). Results are grouped by Index Owner and Indexed Object Owner, and ordered by Index Name.
Function Summary	Lists all non-code-related information about functions. Results are grouped by owner and ordered by name.
Package Summary	Lists all non-code-related information about packages and package bodies. Results are grouped by owner and ordered by name.
Procedure Summary	Lists all information about procedures except for code.

Table 8-1 • Available Reports

Report	Description
Trigger Summary	Lists information about all trigger properties except code. Results are grouped by owner and ordered by name.
Database Link Summary	Lists all information about selected database links.
Snapshot Summary	Lists snapshots by owner and type with creation dates, timestamp, and status.
Object Summary	Lists objects by owner and type with creation dates, timestamp, and status.
User Summary	Lists the database users visible by current user as well as information related to the user.
Object Privileges Summary	Lists object privileges granted to the User.
Procedure Code	Lists information and source code for selected procedures.
Function Code	Lists information and source code for selected functions.
Package Code	Lists information and source code for packages and package bodies.
Table References Object	Lists all objects referenced by the selected tables.
Function References Object	Lists all objects referenced by the selected functions.
Package References Object	Lists all objects referenced by the selected packages.

Table 8-1 • Available Reports

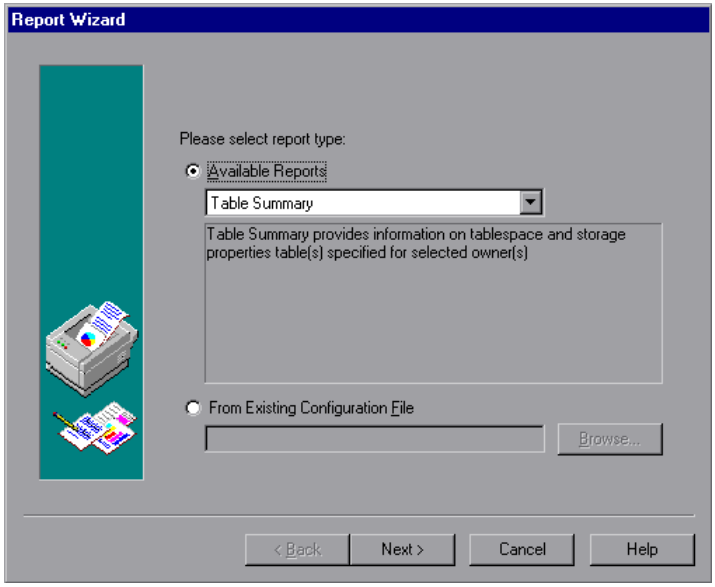
Report	Description
Table Referenced by Object	Lists all objects that reference the specified tables.
Function Referenced by Object	Lists all objects that reference the specified functions.
Package Referenced by Object	Lists all objects that reference the specified packages.
Sequence Referenced by Object	Lists all objects that reference the specified sequences.

Using the Report Wizard

You can use the report wizard to quickly and easily generate reports on a variety of objects.

1 To invoke the Report Wizard, select **Tools, Report Wizard** or click the Report Wizard toolbar button  .

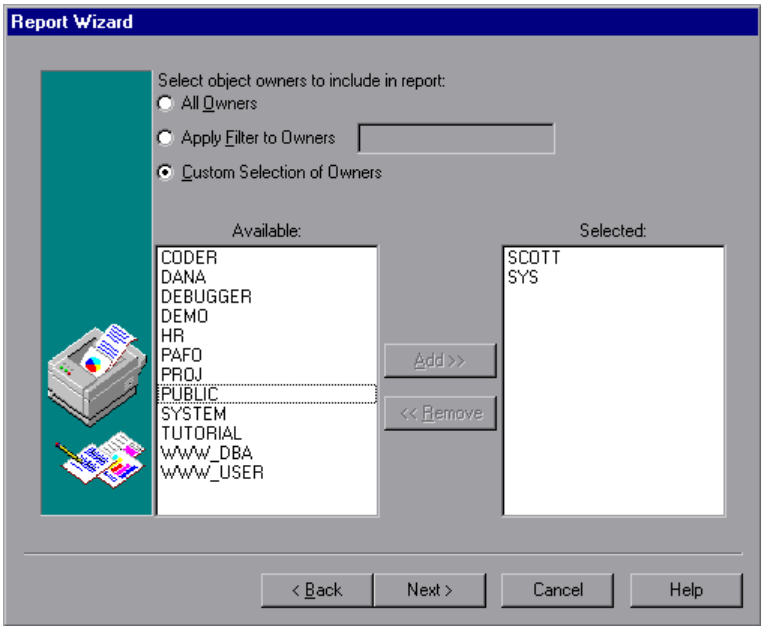
The Report Wizard dialog appears:



You can choose a report type from the **Available Reports** list-box, or an existing report configuration (saved from a prior Report Wizard session).

2 Select a report type and click **Next**.

The Owners page of the Report Wizard displays:

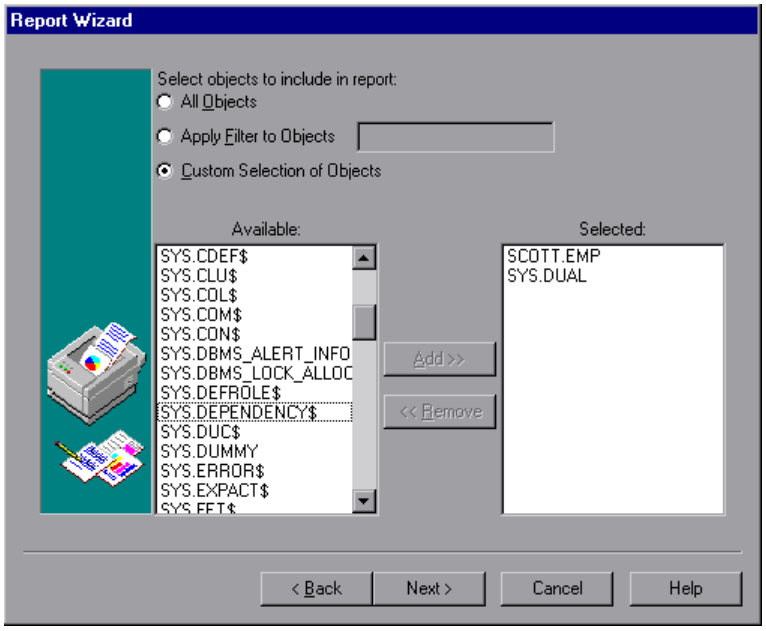


Use this page to specify which owners you want to select objects from for inclusion in the report.

3 Select whether to include all owners, owner names that meet certain criteria, or a custom selection of owners.

- If you choose **All Owners**, you will have the option of including any or all objects on the server in the report, regardless of owner.
- If you choose **Apply Filter to Owners**, you can specify a name filter to limit which owners' objects you can include in the report.
- If you choose **Custom Selection of Owners**, a list-box of available owners displays in the left pane of the dialog. Select one or more owners and click **Add** to move them to the **Selected:** list-box.

4 Click **Next** to display the object selection page:



On this page, specify which objects to include in the report.

Note • **At any point in the Report Wizard, you can click Back to change options on a previous page.**

5 Select whether to include all objects, objects that meet certain name criteria, or a custom selection of objects.

- If you choose **All Objects**, the report will include all objects belonging to the owner(s) specified on the previous page.
- If you choose **Apply Filter to Objects**, you can specify a name filter to limit which objects are included in the report.
- If you choose **Custom Selection of Objects**, a listbox of available objects displays in the left pane of the dialog. Select one or more objects and click **Add** to move them to the **Selected:** listbox.

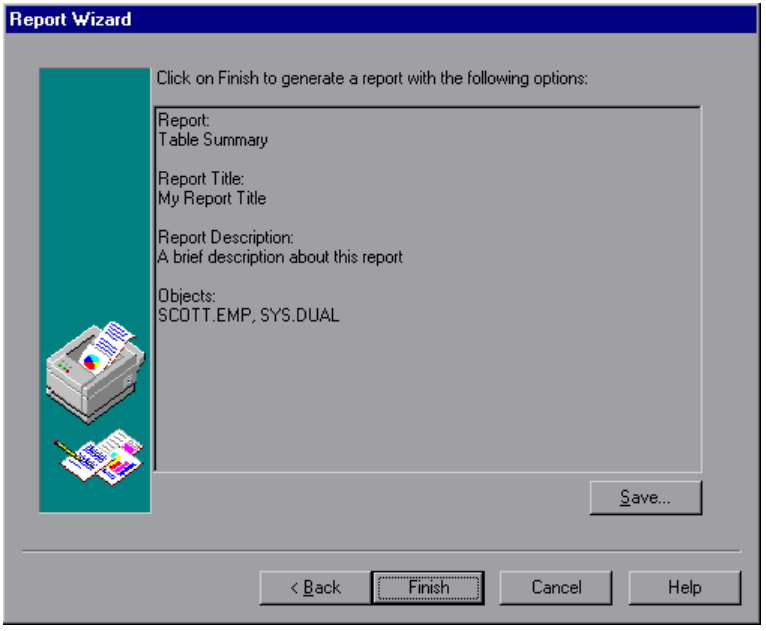
> Note • **If you have specified a name filter, or custom-selected owners on the previous page, the Available listbox will only display objects that meet the criteria.**

6 After you have specified which objects to include, click **Next**.

The Report Title page appears.

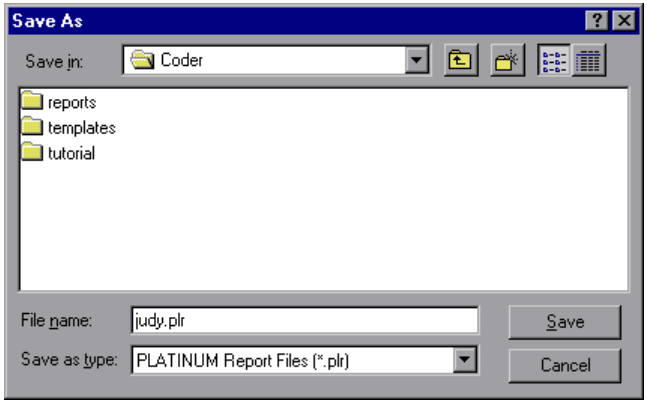
7 Specify a title and description for the report if you wish. If you choose to specify these, they will appear at the top of the generated report. Click **Next**.

A confirmation page appears, displaying the options you have selected for the report.



If you want to change any of the information that is displayed here, click **Back** to navigate backwards through the pages and revise as necessary.

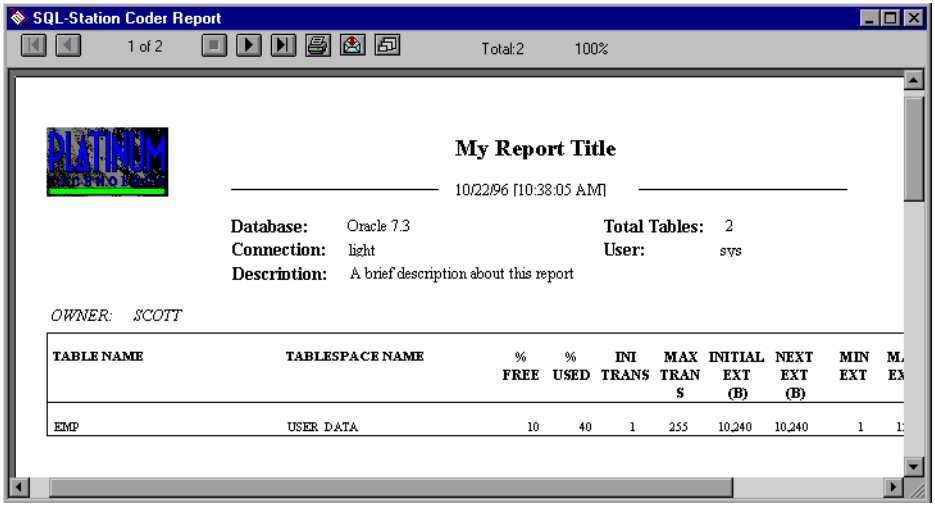
8 If you want to save the report configuration, click **Save**. You are prompted to save the configuration in a **.plr** file (Platinum Report file). If you save a report configuration, you can choose it on the first page of the Report Wizard in future sessions.



Specify a filename and click **Save**. The confirmation page reappears.

9 Verify that the information is correct, and click **Finish** to generate the report.

The report is generated and displayed in a separate SQL-Station Coder Report window.



You can view, print, or export the report using the buttons at the top of the page.

Print. Click this button to print the report.

Zoom. Click this button to view the report at different zoom factors.

Export. Click this button to display the Export dialog, where you can choose from a variety of formats into which to export the report (including Microsoft Word, HTML, and Excel).

Arrow. Click these buttons to move forward and backward through the pages of the report.

9

Setting Coder Preferences

This chapter discusses how to specify preferences for the Coder execution and editing environments.

Setting Coder Preferences 9-2

Catalog Browser Tab 9-3

Edit Window Tab 9-4

Procedure Execution Tab 9-6

General Tab 9-7

Oracle Options tab 9-8

SQL Server Options tab 9-10

Comment Borders tab 9-11

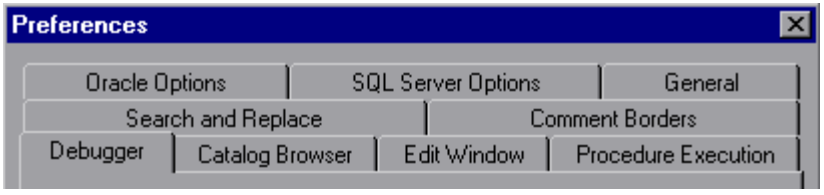
Search and Replace tab 9-12

Debugger tab 9-13

Setting Coder Preferences

Use the Preferences dialog to set preferences for Coder's performance. The Preferences dialog contains the following tabs:

- Catalog Browser
- Edit Window
- Procedure Execution
- General
- Oracle Options
- SQL Server Options
- Search and Replace
- Comment Borders
- Debugger (if SQL-Station Debugger is installed)



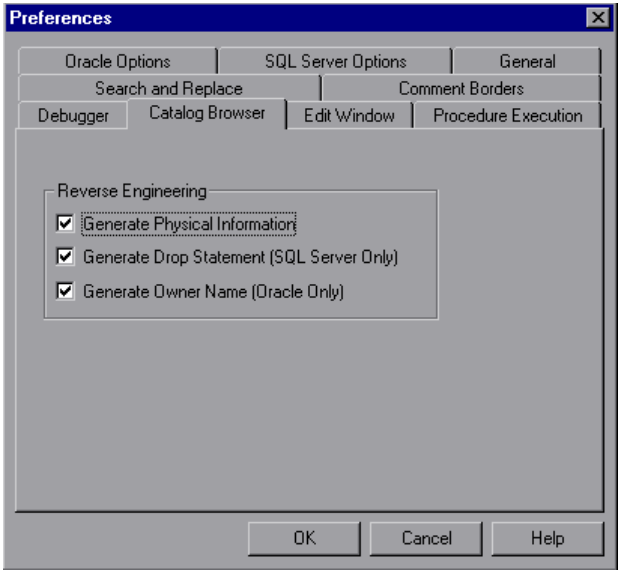
Preferences Dialog tabs

Note • **The Debugger tab is only displayed if SQL-Station Debugger is installed. See the** *SQL-Station Debugger User Guide* **for more information.**

- To invoke the Preferences dialog, select **Edit, Preferences** from the menu or click the Preferences button .

Catalog Browser Tab

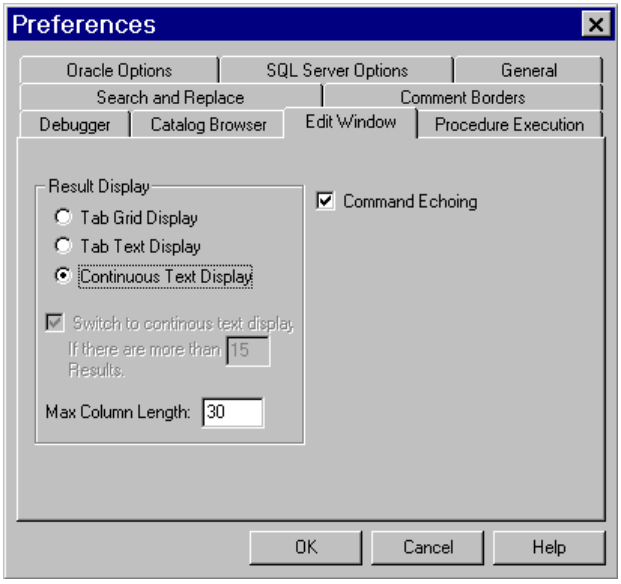
The Catalog Browser tab contains options to configure reverse-engineering.



- Check **Generate Physical Information** to include storage parameters for reverse-engineered objects.
- Check **Generate Drop Statement** to have the reverse-engineered code include a Drop statement before the Create statement (this option applies only to SQL Server objects).
- Check **Generate Owner Name** to include the owner name in the generated code (this option applies only to Oracle objects).

Edit Window Tab

Use the Edit Window tab to specify display characteristics for the Edit Window.



Preferences Dialog (Edit Window Tab)

The Edit Window tab contains the following fields:

Tab Grid Display

Select this to display query results in a grid, on separate result tabs for each query. This is the default.

Tab Text Display

Select this to display query results in text form, on separate result tabs for each query.

Continuous Text Display

Select this to display continuous text results on a single tab.

Switch to continuous text display if there are more than x results

This option is enabled when Tab Text or Tab Grid are selected above. Select this option to automatically switch from tabs to continuous display when the number of queries exceed the specified number.

Max Column Length

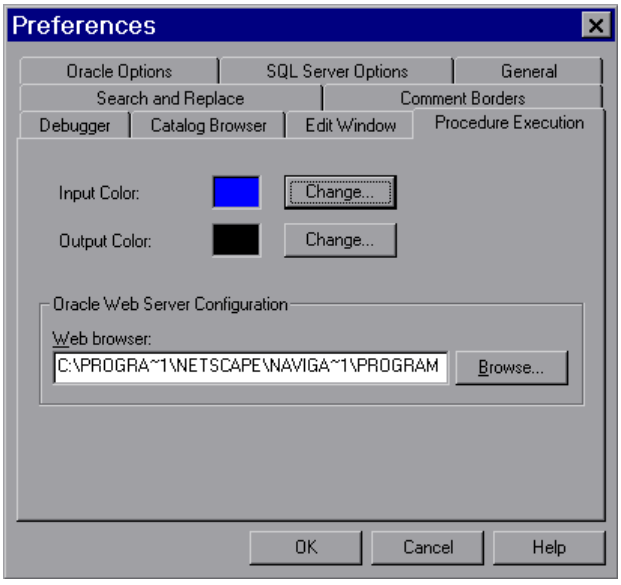
Specify the maximum width for a column in the results display. The default value is 30.

Command Echoing

Check this box to include the SQL statement in the results window.

Procedure Execution Tab

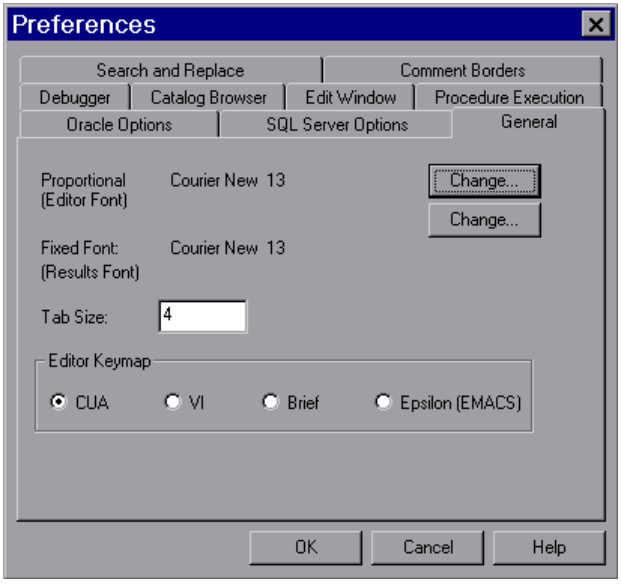
Use the Procedure Execution tab to specify the color-coding for input and output parameters and to specify the location of the default Web Browser for Oracle Web execution.



Preferences Dialog (Procedure Execution Tab)

General Tab

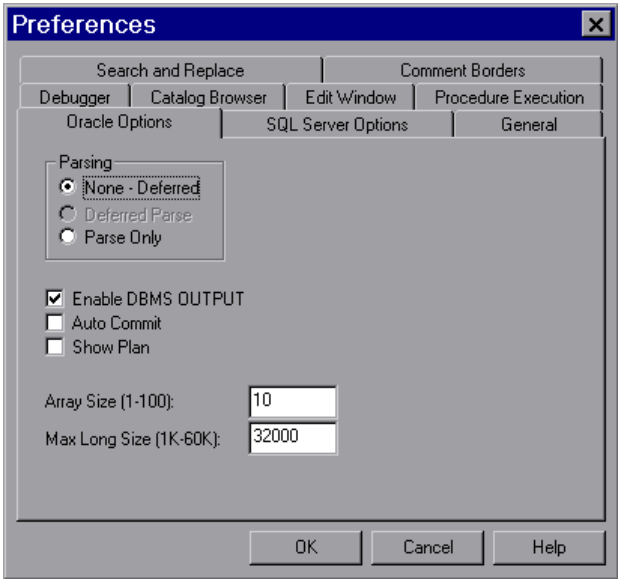
Use the General tab to specify the fonts for the Edit Window and Results Window, and the Codewright editor emulation keymap to use.



Preferences Dialog (General Tab)

Oracle Options tab

Coder provides execution options for each server type. Use the Oracle options tab to configure Oracle statement execution.



Oracle Options tab (Preferences Dialog)

The following Oracle options are available:

Parsing

This option sets Oracle's parsing mode:

- **None-Deferred**: The statement in the Edit Window will be parsed online in a separate step from execution.
- **Deferred Parse**: The statement is not parsed online, but the parse will be combined with the execute in order to enhance performance.

- **Parse Only**: An online parse will be performed with no execution. However, parsing will execute all DDL statements immediately.

DBMS Output

When enabled, includes the contents of any DBMS_OUTPUT commands in the execution results.

AutoCommit

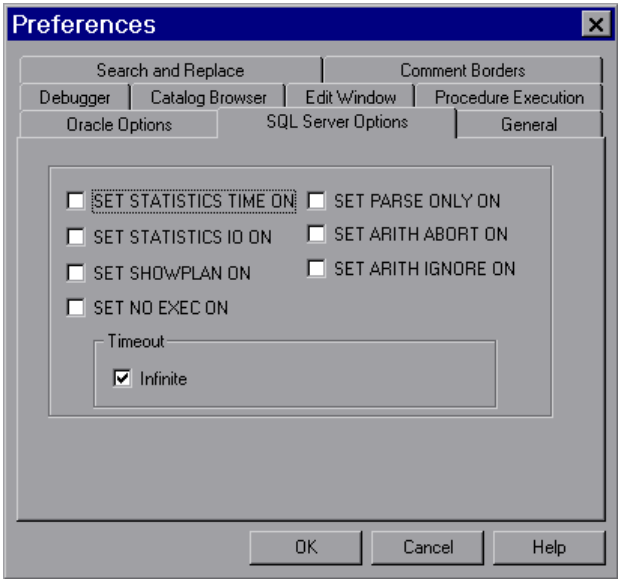
Coder issues a commit after each statement runs in an Edit Window. Rollbacks are not possible when this option is enabled. Auto Commit affects the Edit Window environment, not object execution from the Browser.

ShowPlan

This displays the execution plan of a statement in the Results window.

SQL Server Options tab

Use the SQL Server Options tab to configure SQL Server statement execution.



The following SQL Server options are available:

Statistics

SET STATISTICS TIME ON Displays the CPU time taken for the statement to execute.

SET STATISTICS IO ON Displays the I/O for the statement.

ShowPlan

Returns the query plan.

NoExec

Compiles a statement, but does not execute it. Often used in conjunction with ShowPlan.

Parse Only

Performs a syntax check and returns error messages, but doesn't execute the statement.

Arith Abort

Stop execution when an arithmetic error occurs.

Arith Ignore

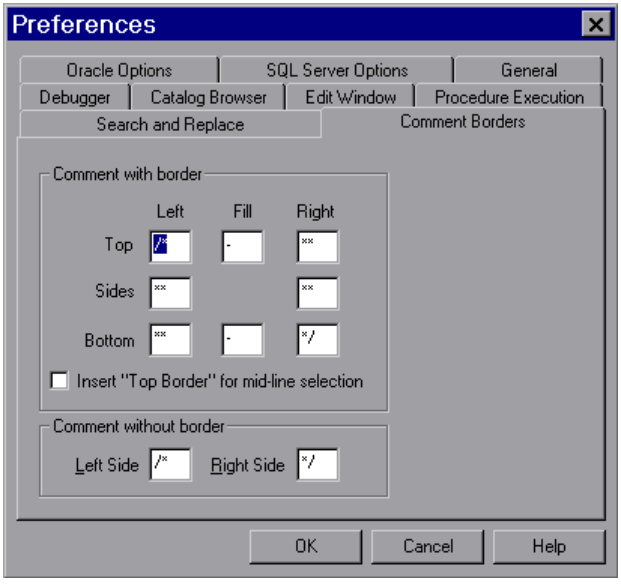
Ignore arithmetic errors and continue execution.

Infinite Timeout

If you check the **Infinite** checkbox, Coder will not timeout.

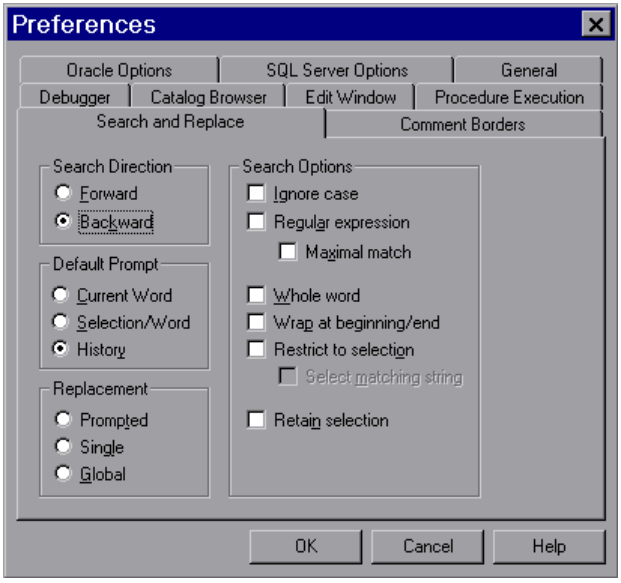
Comment Borders tab

Use the Comment Borders tab to specify preferences for which characters you will use in Coder's automated commenting. Automated commenting is available from the **Text** menu options: **Comment** and **Comment Box**.



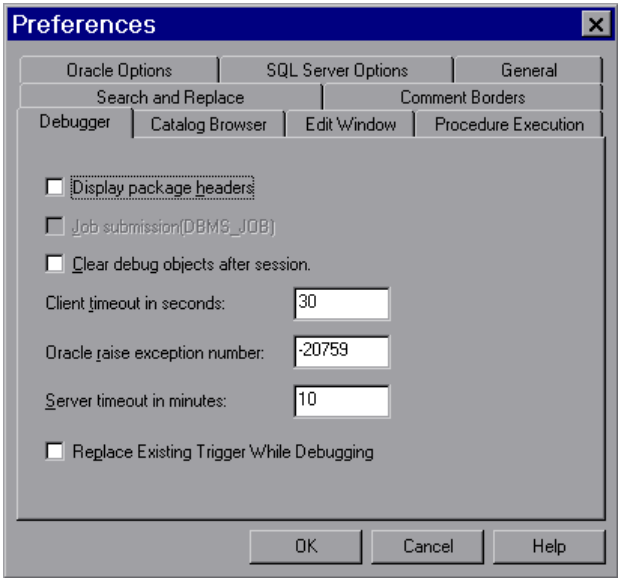
Search and Replace tab

Use the Search and Replace tab to specify preferences for the Edit window Search and Replace function.



Debugger tab

Use this tab to set preferences for debugging sessions. See the *SQL-Station Debugger User Guide* for information about the options on this tab.



Setting Coder Preferences

10

SP Browser

Coder includes an SP Browser utility that facilitates running SQL Server system stored procedures, such as user and database maintenance procedures.

Overview **10-2**

Using the SP Browser **10-2**

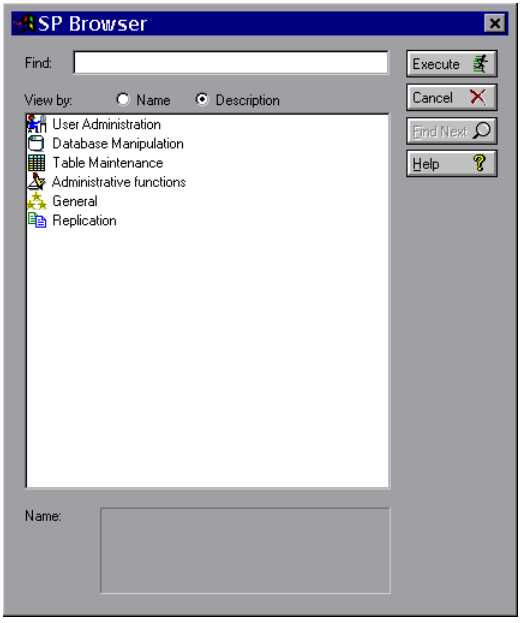
Overview

The SQL Server SP Browser utility lets you easily execute SQL Server system stored procedures to perform administrative tasks. Instead of having to understand the complex naming conventions for the system stored procedures, Coder lets you easily recognize and invoke the procedures.

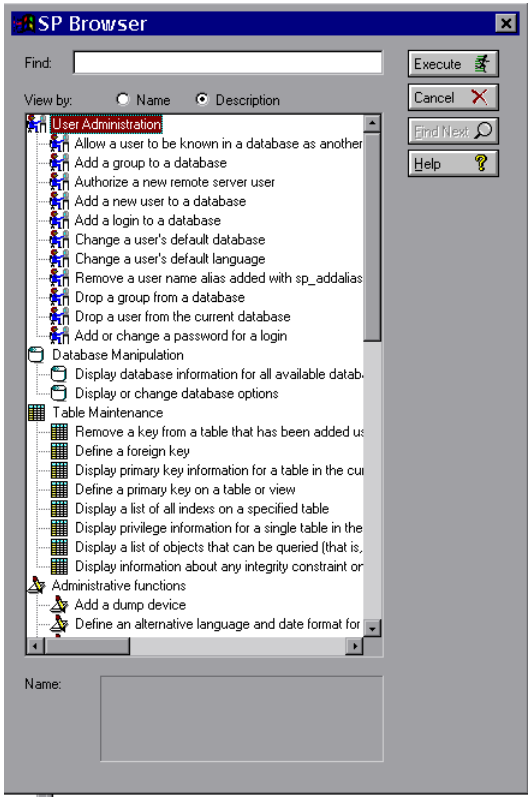
Using the SP Browser

1 Invoke the SP Browser by selecting **Tools, SP Browser**, or clicking the SP Browser toolbar button .

The SP Browser dialog appears:



2 Click the icon for the operation you want to perform. The tree expands to display the administrative task procedures available:



3 Select the procedure you want to execute, and click **Execute**. The procedure invocation facility opens in the Application Workspace. You are prompted to supply any input variables.

4 Supply any input variables, and click the Execute toolbar button 

5 The procedure executes, and results are displayed in the results tabs.

A

Using Coder with Other Tools

This Appendix describes the process of accessing other PLATINUM tools through Coder.

Overview **A-2**

Plan Analyzer for Oracle **A-2**

Invoking Plan Analyzer for Oracle from Coder A-3

Debugger **A-3**

Catalog Browser A-4

File Menu A-4

Overview

If you have purchased and installed the following PLATINUM products, you can access them through Coder:

- Plan Analyzer for Oracle
- Debugger

For detailed information about each product, refer to the documentation provided with the product.

Plan Analyzer for Oracle

PLATINUM's Plan Analyzer for Oracle is a premier Oracle optimization tool. You can use Plan Analyzer for Oracle to do the following:

- Optimize SQL statements (SELECT, UPDATE, INSERT, and DELETE)
- Capture SQL from the server
- Compare optimization plans, and generate reports
- Watch Server Statistics

If you have purchased and installed Plan Analyzer for Oracle, you can use it from within Coder to tune SQL statements of the following types:

- SELECT
- UPDATE
- INSERT
- DELETE

For detailed information about installing and using Plan Analyzer for Oracle, consult the *Plan Analyzer for Oracle User Guide*, provided with Plan Analyzer for Oracle.

Invoking Plan Analyzer for Oracle from Coder

If you have installed Plan Analyzer for Oracle, you can invoke it from an active Edit Window. When an Edit Window contains a SELECT, UPDATE, INSERT, or DELETE statement, the Plan Analyze button  is enabled on the toolbar, and the **Plan Analyzer** option is enabled on the **Tools** menu. Follow these steps to invoke Plan Analyzer for Oracle on a SQL statement:

1. With a SQL statement in the active Edit Window, select **Tools, Plan Analyzer** or click the Plan Analyze toolbar button.

   Coder starts up Plan Analyzer for Oracle and copies the SQL statement from the Edit Window into the Plan Analyzer SQL Window.

2. Use Plan Analyzer to do any of the following:

   - generate plans or server statistics for the SQL statement
   - modify and save the SQL and plans to the Plan Analyzer repository
   - save the SQL to a .sql file
   - copy and paste the statement back into the Coder Edit Window

3. When you want to return to Coder, task over (ALT + TAB), or close Plan Analyzer and you are automatically returned.

Debugger

PLATINUM's Debugger is a state-of-the-art debugging and execution tool for Oracle server-side database objects. You can use Debugger to do the following:

- examine and compare the structure of database objects
- set breakpoints
- see the flow of execution
- see changes in the values of variables as you move through the code

- step into, over, or out of any program module

If you have Debugger installed on your system, you can invoke it from the Catalog Browser or **File** menu. To start Debugger from Coder, follow these steps:

Catalog Browser

1. Navigate through the Catalog Browser tree (Oracle connection) to a procedure, function, package, or trigger that you want to debug.
2. Highlight the object. The Debug toolbar button is enabled.
3. Click the Debug button to begin the debugging process.

Refer to the Debugger User Guide for information about Debugger.

File Menu

Select **File, File Debug** from the menu. Use the Open dialog to select a file that contains PL/SQL.

Refer to the *Debugger User Guide* for information about Debugger.

Index

.plr files 8-9
.sql files 5-4
.tpl files 5-16

A

active connection 2-4, 2-8, 4-16, 6-9
adding a connection 4-16
adding groups 2-10
API Assistance option 5-12
Application Workspace 2-4
arith abort 9-10, 9-11
arith ignore 9-10, 9-11
associated connection
  Edit Window 5-5
Auto Commit 9-8
AutoCommit 9-9

B

Brace Matching 5-4, 5-11
By Name filter 4-13
By Owner filter 4-5, 4-13
By Type filter 4-5, 4-13

C

Catalog Browser 2-3, 4-1–4-23
  as window 4-4
  code icon 5-4
  contents 4-4
  delete table data 4-13
  filters 4-5, 4-10, 4-13
  icons 4-6, 4-7
  navigating 4-10
  object categories 4-10
  opening 4-4
  Oracle objects 4-4
  overview 2-12, 4-3
  procedures 7-2
  reverse-engineering from 2-3
  SQL Server objects 4-4
Catalog Browser tab
  Preferences dialog 9-3
client-side installa 1-3
client-side installation 1-2

code icon 4-11, 4-22, 5-4, 5-5
code templates 5-13–5-16
Coder
  installation 1-2
  logging in 2-6
  main window 2-4
  overview 2-2
  typical workflow 2-2
Codewright
  online help 5-7, 5-11
Codewright editor 5-2, 5-7
color-coding
  and filename extensions 5-11
  SQL 5-11
Command History combo-box 5-2
Comment Borders tab
  Preferences 9-11
  preferences 9-2
comments 9-11
commit 9-8
Compare utility 6-5–6-6
connect string 2-7
connecting 2-7, 5-3
connection
  active 2-4, 2-8, 4-16, 5-2, 5-5, 6-9
  additional 2-8, 4-16
  re-establishing 2-9
  switching between 2-8, 4-16
  terminating 2-9
  working without 5-3
continuous text display
  results window 9-5
current connection 4-16, 5-2
  and Edit Window 5-5
  and Find Objects utility 6-3

D

data
  delete 4-13
Data icon 4-11
database
  SQL Server 4-4
Database menu
  Open Connection option 4-16
DBMS output 7-2, 9-8, 9-9
DDL
  generation 6-12
Debugger A-2
debugging Oracle procedures A-3
deferred parse 9-8
delete data 4-13
disconnecting 2-9
DML
  generating 6-14
drop object 4-12, 4-17

E

Edit menu
  Find option 5-9
Edit Window 2-3, 5-2
  brace matching 5-11
  Codewright features 5-2, 5-7
  connection 4-16, 5-5, 6-9
  drag and drop table objects 6-14
  editor emulation 5-7, 5-8
  entering SQL 5-4
  opening 5-3
  overview 2-13, 5-2
  preferences 5-6, 9-2
  right-click popup menu 5-4
  syntax highlighting 5-11

templates 5-13–5-16
toolbar button 5-3
version control 5-10
Edit Window tab
Preferences dialog 9-4
editor emulation
preferences 9-2, 9-7
execute function 4-12
execute procedure 4-12, 4-23
executing
procedures and functions 7-2
executing SQL Server system stored procedures 10-2
exporting reports 8-10

F

File menu
New option 5-3
Open option 5-4
filename extensions
.plr 8-9
.sql 5-4, 5-11
.tpl 5-16
filter
By Name 4-13
By Owner 4-5, 4-13
By Type 4-5, 4-13
filters 4-13
Catalog Browser 4-5
find object utility 6-3
Find option 5-9
font preferences 9-7
function
execute 4-12
looking up 5-6, 6-14
recompile 4-12
function execution 7-2

G

General tab
Preferences dialog 9-7
generate insert script 4-12, 4-19
generate package 4-13
generate procedure 4-12, 4-19
Generate SQL Script 4-22, 5-5, 6-7–6-9
generate statement 4-12, 4-18
groups
adding 2-7, 2-10
removing 2-11

H

Help menu
API Assistance option 5-12
History button 2-9
history of logins 2-7, 2-9

I

icon
code 4-11
Data 4-11
Info 4-11
Referenced By 4-11
References 4-11
icon, Privileges 4-11
icons
Catalog Browser 4-6, 4-7
infinite timeout 9-11
Info icon 4-11
insert script
generating 4-12, 4-19

installation 1-2
client-side 1-2, 1-3
server-side 1-2, 1-4
server-side (Oracle) 1-5
server-side (SQL Server) 1-8
system requirements 1-2

L

login
history 2-7
Login dialog 2-6, 2-7
History button 2-9
Restore button 2-9
Server Types field 2-8
Lookup utility 6-12

M

main window 2-4
menu
right-click 2-12, 4-12

N

NoExec 9-10, 9-11
none-deferred
parsing 9-8

O

object categories 4-4
Object Lookup utility 5-6, 6-12–6-16
object templates 5-13–5-16
objects
dropping 4-17
looking up information about 6-12
viewing source code 4-22
Oracle
SQL tuning tool A-2
Oracle execution options 9-2, 9-8
Oracle objects
Catalog Browser 4-4, 4-6
Oracle Options tab
Preferences dialog 9-8

P

package
generate 4-13
looking up 5-6, 6-15
parse only 9-8, 9-9, 9-10, 9-11
parsing 9-8, 9-10
PL/SQL code templates 5-13–5-16
Plan Analyzer for Oracle A-2
preferences 9-2–9-13
Edit Window 5-6
setting 2-12, 5-6, 7-5, 9-2
Preferences dialog 2-12, 9-2
Catalog Browser tab 9-3
Comment Borders tab 9-11
Edit Window tab 9-4
General tab 5-8, 9-7
Oracle Options tab 9-8
Procedure Execution tab 9-6
Search and Replace tab 9-12
SQL Server Options tab 9-10
Privileges icon 4-11
procedure
execute 4-12, 10-3
generate 4-12, 4-19
input variables 7-4
looking up 6-16
recompile 4-12
procedure execution 4-23, 7-2
preferences 7-5, 9-2, 9-6

SQL Server System 7-5
Procedure Execution tab
Preferences dialog 9-6
Procedure Execution window 7-4
procedures
debugging A-3

Q

Quick Search option 5-4

R

recompile function 4-12
recompile procedure 4-12
recompile trigger 4-12
re-establishing a connection 2-9
Referenced By icon 4-11
References icon 4-11
removing groups 2-11
report wizard 8-2–8-10
reports 8-2
exporting 8-10
reverse-engineering 2-3, 3-14, 4-11, 4-22, 5-4, 6-7
preferences 9-3
right-click popup menu 2-12, 4-12, 5-4
Delete Data option 4-13
Drop option 4-12
Execute option 4-12
Generate Insert Script option 4-12
Generate Package option 4-13
Generate Procedure option 4-12
Generate Statement option 4-12
Recompile option 4-12
Update Statistics option 4-12
running SQL code 5-6

S

schema
Oracle 4-4
Scripting tool 6-7
Search 5-9
Search and Replace 5-9
Search and Replace preferences 9-2
Search and Replace tab
Preferences dialog 9-12
server
connecting to 2-6
establishing new connection 2-7
switching to different 2-8
Server Messages Pane 2-4
server types 2-8
server, connecting to 2-7, 2-8
server-side installation 1-2, 1-4
ShowPlan 9-8, 9-9, 9-10, 9-11
SP Browser 10-2
Spell Checking 5-4, 5-12
SQL
color-coding 5-11
entering in Edit Window 5-4
running 5-6
tuning with Plan Analyzer A-2
SQL menu
Execute option 5-6
SQL Scripting utility 4-22, 5-5, 6-7
SQL Server execution options 9-2, 9-10
SQL Server objects
Catalog Browser 4-4, 4-7
SQL Server Options tab
Preferences dialog 9-10
SQL Server system stored procedures 10-2
SQL Wizards 5-5, 6-10–6-12

statement
generate 4-12, 4-18
statistics 9-10
update 4-12
stored procedure execution 4-23
switching connections 2-8, 4-16
syntax highlighting 5-11
system requirements 1-2

T

table
delete data 4-13
generate insert script 4-12, 4-19
generate statement 4-18
looking up 5-6, 6-13
templates
PL/SQL 5-13–5-16
Transact-SQL 5-13–5-16
timeout
infinite 9-11
toolbars 2-5
Tools menu 4-23, 6-2
Catalog Browser option 4-4
Compare option 6-5
Edit Window option 5-3
Find Object option 6-3
Generate SQL Script option 4-22, 5-5, 6-7
Lookup option 5-6
Object Lookup 6-12
Plan Analyze option A-3
Procedure Execution option 7-2
Report Wizard option 8-2
SP Browser option 10-2
Spell Checking option 5-12
SQL Wizards option 6-11
Transact-SQL code templates 5-13–5-16
trigger
recompile 4-12

U

update statistics (SQL Server) 4-12
user accounts 4-4

V

VCS 5-10
Version Control 5-10
API Assistance option 5-4