AT&T 585-350-914
Issue 1
Comcode 407298645

SQL*MENU® USER'S GUIDE AND REFERENCE

VERSION 5.0

ORACLE®

The Relational Database Management System

SQL*Menu User's Guide and Reference
Version 5.0
Part No. 3303-V5.0 0490

Contributing Author: Lefty Leverenz
Contributors Eric Bond, Robert Giljum, Linda Schlueter, Chris Schock, Herman Slange

---

Acknowlegdement: The contributing authors for the SQL*Menu Online Help System are Linda Schlueter (SQL*Menu help screens) and Gerry Glassmeyer (SQL and PL/SQL help screens).

PREFACE

The *SQL*Menu User's Guide and Reference* explains how to use SQL*Menu Version 5.0. It shows you how to run menu-based software applications, how to design menus and menu structures, and how to administer SQL*Menu.

This preface describes the organization of this book and how you can use it most effectively.

Structure of This Book

This book provides comprehensive information about SQL*Menu Version 5.0. Reference information (descriptions and explanations) appears separately from procedural information ("how to do it" sections), so you do not have to read through information in which you are not currently interested.

Major Sections of This Book

The *SQL*Menu User's Guide and Reference* consists of the following major sections, or parts:

- Part I: About SQL*Menu
- Part II: SQL*Menu Operator's Guide
- Part III: SQL*Menu Designer's Guide
- Part IV: SQL*Menu Reference
- Part V: Appendixes

See the table of contents for a chapter-level overview of the major parts of this book.

How to Use This Book

This book contains information for three different audiences: application operators, application designers, and database administrators.

Operators use SQL*Menu to work quickly and easily with various software applications. *Designers* use SQL*Menu to build custom menu applications that operators can use for virtually any data-processing purpose. *Database administrators* (DBAs) maintain the software and enroll users for SQL*Menu and related products.

Part I, "About SQL*Menu," should be read by everyone who uses this book.

Part II, "SQL*Menu Operator's Guide," is primarily for operators of SQL*Menu applications who only need to know how to run an application.

Part III, "SQL*Menu Designer's Guide," and Part IV, "SQL*Menu Reference;" are primarily for designers of menu applications. Designers should also read Parts I and II for basic information about using SQL*Menu.

The online help system can also assist designers in learning how to design SQL*Menu applications. If you need more explanation than the , online help system provides, use this book's index to find information about related topics.

Chapter 5, "Design Operations: in Part III and **Chapter 8, "SQL*Menu Components and Administration:** in Part IV contain information for database administrators. **Appendix B, "Error Messages,**" explains the error messages that might appear while running or designing a menu application.

Typographic Conventions

This book uses typographic conventions to distinguish important elements from the body text of the manual. You should be familiar with these conventions.

Function Keys

SQL*Menu function keys are represented by the name of the SQL*Menu function enclosed in square brackets (e.g., [Next Field]). You can refer to a keypad diagram or the SQL*Menu Show Keys screen (online) for a mapping of these function keys to physical keys on your keyboard.

Screen Messages

Hint messages and error messages appear in monotype. For example:

```
MNU-20011:  Valid menu name must be entered.
```

Command Syntax

Commands and parts of commands appear in monotype. For example:

```
HOST ( {' command_ string' I ' variable' } [, NOSCREEN] ) ;
```

Parts of commands are set off from each other by the following typographic conventions:

Convention	*Explanation*
`regular monotype`	Indicates a part of the command that you must enter exactly as shown.
`underlined monotype`	Indicates a default parameter. If you indicate no parameter in a parameter set, SQL*Menu applies the default parameter.
`italic monotype`	Indicates a variable. You must substitute an appropriate value when you enter the command.
`|(vertical bar)`	Separates alternative items.
`{ } (braces)`	Encloses a set of alternative items from which you are required to choose.
`[ ] (brackets)`	Encloses an optional item or a set of alternative items from which you have the option to choose.

Your Comments Are Welcome

As we write, revise, and evaluate this manual, we value your comments. Please use the Reader's Comment Form at the back of this document to tell us what you liked and disliked about this book. Or, you may contact us at the following address:

SQL*Menu Product Manager
Oracle Corporation
20 Davis Drive
Belmont, California 94002
(415) 598-8000

CONTENTS

PREFACE

PART 1 **ABOUT SQL*MENU**

Chapter 1 **SQL*Menu Overview** **1-1**

What Is SQL*Menu? 1-2

Software Integration through SQL*Menu 1-2

SQL*Menu System Components 1-3

Features and Benefits of SQL*Menu 1-3

The Dynamic Menu Approach 1-5

PL/SQL: A Procedural Extension to SQL 1-5

A Sample Menu Application 1-5

Menu Appearance 1-6

Menu Tree for the Sample Application 1-7

Requirements for Using SQL*Menu 1-8

Hardware 1-8

Software 1-8

Authorization to Use SQL*Menu 1-9

The Database Administrator 1-9

Installing SQL*Menu 1-10

PART II **SQL*MENU OPERATOR'S GUIDE**

Chapter 2 **Running a Menu Application** **2-1**
Starting and Exiting SQL*Menu (Run Menu) 2-2
Logging Into SQL*Menu (Run Menu) 2-2
Exiting SQL*Menu (Run Menu) 2-3
Selecting Menu Item 2-4
Selecting by Letter 2-5
Selecting by Cursor Movement Keys 2-6
Entering Parameter Values 2-8
Getting Help Information 2-10
Context 2-10
The Message Line 2-10
Function Key Help 2-11
The Application Help System 2-11
Using the Background Menu 2-12
Using Operating System Commands 2-14

Chapter 3 **Menu Styles and Navigation** **3-1**
Navigating in SQL*Menu (Run Menu) 3-2
Specifying a Menu Display Style 3-3
Using Pull-Down and Bar Style Menus 3-3
Pull-Down Style Menus 3-4
Bar Style Menus 3-5
Using Full-Screen Menus 3-6
Navigating and Selecting Items in Full-Screen Menus 3-7
The Where Option: Menu Names and Locations 3-9

Chapter 4 **Function Keys for Menu Operators** **4-1**
Function Keys for SQL*Menu (Run Menu) 4-2
Function Key Mappings 4-2
Function Key Descriptions 4-2

Functional Groupings for Operators 4-7
Cursor Movement Functions 4-8
Editing Functions 4-8
General Functions 4-8
User Assistance Functions 4-8

PART III **SQL*MENU DESIGNER'S GUIDE**

Chapter 5 **Design Operation** **5-1**
Overview of Menu Design 5-2
Starting and Quitting SQL*Menu (Design) 5-3
Logging In to SQL*Menu (Design) 5-3
Quitting SQL*Menu (Design) 5-4
Basic Design Operations 5-6
Getting Help Information 5-6
Creating a New Application 5-8
Opening a Menu Application 5-9
Modifying an Application Definition 5-10
Defining a Menu 5-12
Defining a Menu Item 5-14
Defining a Substitution Parameter 5-17
Defining a PL/SQL Procedure 5-21
Copying or Referencing an Object 5-23
Saving an Application 5-24
Generating an Application 5-25
Executing a Menu Application 5-26
printing Application Information 5-27
Setting Options in SQL*Menu (Design) 5-29
User Preference File 5-31

Application Maintenance Operations 5-33
Copying an Application 5-33
Renaming an Application 5-35
Deleting an Application 5-36
Unloading (Exporting) an Application 5-37
Security Operations 5-38
Granting Access to SQL*Menu 5-39
Revoking Access to SQL*Menu 5-41
Creating Roles 5-42
Assigning Users to Roles 5-45

Chapter 6 **Elements in the Design Interface 6-1**
Context in the Design Interface 6-2
Hierarchy and Context 6-2
Function Keys and Context 6-2
Menu Elements 6-3
Menu 6-3
Menu Item 6-4
Message Line 6-4
Status Line 6-4
Data-Entry Elements 6-5
Alert 6-5
Button or Radio Button 6-6
Check BOX 6-8
Dialog Box 6-8
Field 6-9
Form 6-10
List of Values 6-11
Scroll Region 6-13
Spread Table 6-14

	The Online Help System	6-17
	Displaying Context-Sensitive Help	6-17
	Navigating in the Online Help System	6-18
Chapter 7	Function Keys for Designers	7-1
	Function Keys for SQL*Menu (Design)	7-2
	Function Key Mappings	7-2
	Function Key Descriptions	7-2
	Functional Groupings for Designers	7-12
	Cursor Movement Functions	7-12
	Editing Functions	7-12
	General Functions	7-12
	User Assistance Functions	7-12
PART IV	**SQL*MENU REFERENCE**	
Chapter 8	**SQL*Menu Objects**	**8-1**
	Object Hierarchy	8-2
	Object-Naming Conventions	8-3
	File Names	8-3
	Reserved Words	8-3
	Applications	8-5
	Defining Applications	8-5
	Application Characteristics	8-6
	Menus	8-9
	Defining Menus	8-9
	Menu Characteristics	8-11
	Menu Items	8-12
	Defining Menu Items	8-12
	Menu Item Characteristics	8-14
	SQL*Menu Command Types	8-16

Substitution Parameters 8-21
Defining Substitution Parameters 8-21
parameter Characteristics 8-23
Using Substitution Parameters 8-25
Standard Substitution Parameters 8-25
Parameter Processing 8-26
Procedures 8-26
Defining Procedures 8-26
Procdure Characteristics 8-28
Using Procedures 8-28

Chapter 9 PL/SQL 9-1
PL/SQL Overview 9-2
PL/SQL Blocks in SQL*Menu 9-3
Anonymous Blocks 9-4
Procedure Blocks 9-4
Referencing Objects in PL/SQL 9-6
Reference Syntax 9-6
Invoking Packaged Procedures 9-6
PL/SQL Variables 9-6
Restrictions on Referencing Objects 9-8
PL/SQL Compilation 9-8
Compiling at Design Time 9-9
Compiling at Generation Time 9-9
Compilation Error Messages 9-9
Executable Statement Failure 9-10
SQL Statement Failure 9-11
Exception Handling 9-11

Chapter 10 **PL/SQL Packaged Procedures** **10-1**
Packaged Procedure Overview 10-2
Packaged Procedure Failure 10-2
SQL*Menu States 10-2
Testing Packaged Procedure Execution 10-3
Packaged Procedure Arguments 10-4
SQL*Menu Packaged Procedures 10-5
SQL*Forms Packaged Procedures 10-15

Chapter 11 **SQL*Menu Macros** **11-1**
Macro Overview 11-2
Macro Syntax 11-2
Macros, Packaged Procedures, and Function Keys 11-2
Macro Failure 11-4
Macros without Packaged Procedures 11-5
Macros with Arguments 11-6

Chapter 12 **SQL*Menu Components and Administration** **12-1**
Automatic Login 12-2
Remote Login 12-2
SQL*Menu System Components 12-3
SQL*Menu (Design) 12-4
SQL*Menu (Run Menu 12-4
SQL*Menu (Generate) 12-6
SQL*Menu (Document) 12-8
SQL*Menu Constraints 12-9
Terminal Definitions 12-9
Views of the SQL*Menu Base Tables 12-10

PART V **APPENDIXES**

Appendix A **Upgrading SQL*Menu** **A-1**
Upgrading Menu Applications A-2
Upgrading One Application A-2
Security System A-2
National Language Support A-3

Appendix B **Error Messages** **B-1**
SQL*Menu (Run Menu) Error Messages B-3
SQL*Menu (Document) Error Messages B-12
SQL*Menu (Generate) Error Messages B-14
SQL*Menu (Design) Validation Error Messages B-19
PL/SQL Compilation Error Messages B-31

Glossary

Index

PART

I

About SQL*Menu

CHAPTER

1 SQL*MENU OVERVIEW

This chapter introduces SQL*Menu and explains the benefits that SQL*Menu offers to the designers and operators of menu applications. It also describes the components of SQL*Menu and the authorization you need to use the different components.

This chapter covers the following topics:

- what is SQL*Menu?
- features and benefits of SQL*Menu
- a sample menu application
- requirements for using SQL*Menu

What Is SQL* Menu?

SQL*Menu is a productivity tool that provides a single menu interface for running multiple data-processing tools. SQL*Menu has three categories of users:

- *Operators* use SQL*Menu to run software applications.
- . *Designers* use SQL*Menu to create menu applications.
- *SQL*Menu administrators* (SQL*Menu DBAs) maintain SQL*Menu and enroll operators and designers.

A *menu* is a list of choices, or *menu items,* that operators can select at run time to specify their next action. Menu items can call other menus, execute commands, or run programs.

A *menu application* is a set of interconnected menus from which operators can carry out their data processing needs. The structure of a menu application is represented by a menu tree.

A menu tree has a root menu, or *main menu,* connected to other menus, called submenus. A submenu is a menu that is called by an item in another menu, the parent menu. Any submenu can also be a parent menu if its items call other submenus.

Software Integration through SQL*Menu

A SQL*Menu application acts as an "umbrella" under which you can integrate many different software products to provide a single solution for your business needs.

If you have several menu applications for distinct business purposes, SQL*Menu lets operators move from one application to another easily, without leaving SQL*Menu. For example, a manager reviewing the data in an order-entry application can quickly switch to an inventory application to verify the stock on hand.

SQL*Menu can be used with Oracle software products and with other software products that run on your computer's operating system. Any valid operating system command can be invoked in a SQL*Menu application.

SQL*Menu System Components

SQL*Menu Version 5.0 contains several system components:

SQL*Menu (Design)	Designers use this component to define and modify menu applications, and SQL*Menu administrators use it to maintain the security system.
SQL*Menu (Generate)	Designers use this component to create a library file from an application defined in SQL*Menu (Design).
SQL*Menu (Document)	Designers use this component to print or store information about an application defined in SQL*Menu (Design).
SQL*Menu (Run Menu)	Operators use this component to run a menu application created in SQL*Menu (Design) and generated by SQL*Menu (Generate).

Menu items in the interface of SQL*Menu (Design) can call the other system components, and any SQL*Menu system component can be called directly from the command line of your operating system.

For full descriptions of the system components, see "SQL*Menu System Components" in Chapter 12.

Features and Benefits of SQL*Menu

A menu interface can benefit software applications in the following ways:

- display all the choices available to an operator
- reduce the amount of technical knowledge operators need
- add structure to an application, making it easier to learn, use, and maintain
- minimize typing errors
- improve security by restricting the number of commands that an operator can enter
- reduce training and support costs for an organization

SQL*Menu achieves these benefits through the following special features

Choice of menu display styles. A menu application can appear in the display style most appropriate for that application:

- pull-down style
- bar style
- full-screen style

Hierarchical and networked linking of menus. Operators can move from one menu to another by navigating up and down the menu tree. If the application runs in the full-screen display style, operators can move directly from one menu or menu item to another.

Functions as menu items. Designers can create menu items that execute one or more SQL*Menu functions. Packaged procedures provide simple commands and PL/SQL provides a powerful command language.

Run-time substitution of parameter values. SQL*Menu can prompt the operator for information at run time and incorporate the information into commands executed by the menu.

Menu-driven execution of operating system commands. Operators need not learn complex operating system commands, because menu items can be designed to accomplish specific tasks automatically.

Execution of user-entered operating system commands. Operators can enter operating system commands from within the menu structure.

Background menus. A background menu, which is available from any part of the menu hierarchy, can provide "accelerator keys" for frequently used commands.

Application security and dynamic menus. SQL*Menu maintains security at the application level and at the level of individual menu items. Access depends on the operator's membership in security roles. Optionally, SQL*Menu either displays or conceals items that the operator cannot use.

Integration with SQL*Forms Version 3.0. A SQL*Menu application can invoke one or more forms created by SQL*Forms. Alternatively, SQL*Forms can invoke SQL*Menu, displaying a menu within a form.

The Dynamic Menu Approach

SQL*Menu can generate *dynamic menus* whose content changes depending upon the access privileges of the operator running the menu application. For example, one operator may be able to use only three items on a menu, while another operator maybe able to use all ten items on the same menu.

Menu items that the current operator can use appear in bold characters on the screen. SQL*Menu can either conceal the unavailable menu items or display them with non-bold characters (or a specific color). The designer of a menu application decides item-by-item whether to conceal or display unavailable items.

Authorization to use menu items is granted to various privilege categories, called *roles.* A role accesses a given set of menu items. Each user can be a member of one or more roles.

When you run a SQL*Menu application, your role membership determines which items are available to you and which items are suppressed (or displayed as unavailable). If you belong to several roles, you will have access to all items available to your combined roles.

PL/SQL A Procedural Extension to SQL

PL/SQL is a programming language that extends SQL by adding procedural capabilities such as looping, branching, and error handling.

In SQL*Menu applications, a menu item can invoke a *PL/SQL block,* which is the basic programming unit in PL/SQL. APL/SQL block can contain any number of procedural statements and SQL statements.

For complete information about PL/SQL, see the *PL/SQL User's Guide and Reference.* For information about SQL, see the *SQL Language Reference Manual.*

A Sample Menu Application

This section introduces a sample application called the Orders Application, which is used as an example throughout this book to demonstrate SQL*Menu principles and operations.

The Orders Application calls a sample form so operators can enter, update, and query information about sales orders for the Summit Sporting Goods company. This sample form, also named ORDERS, is a copy of the form created in the *SQL*Forms Designer's Tutorial. The* Orders Application also runs several types of reports about sales orders and customers, and provides utility functions for the operators' convenience.

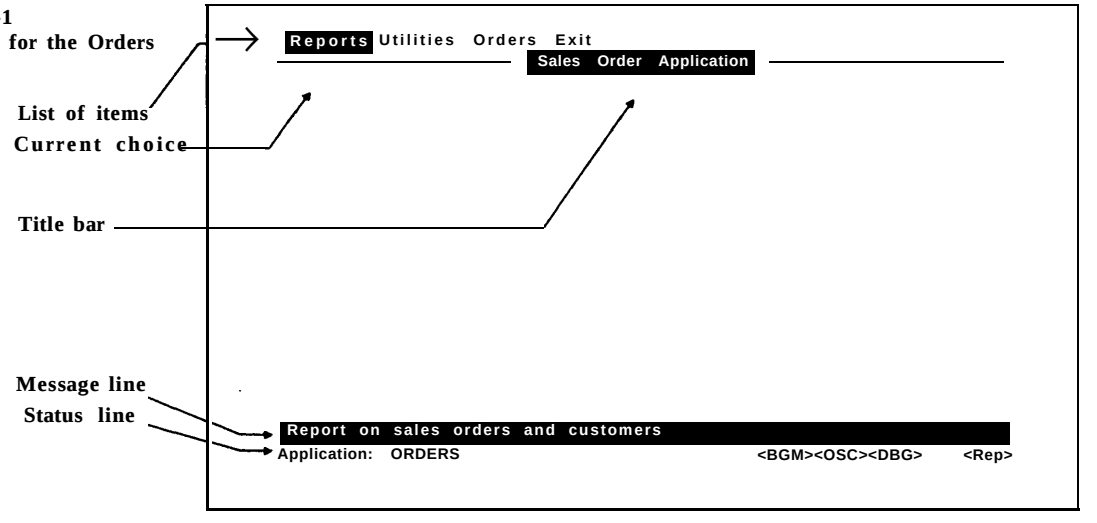
Another sample application, named Ordform, is similar to the Orders Application except that it is called by its sample form, named ORDFORM, instead of calling the form from the menu application. The sample applications are examples of integrating SQL*Forms Version 3.0 with SQL*Menu Version 5.0.

The Orders and Ordform Applications are also available as online practice applications; see your DBA for instructions on accessing them.

Menu Appearance

Figure 1-1 shows a menu from the Orders Application in the standard pull-down menu display style. For illustrations of the other menu display styles, see Chapter 3, "Menu Styles and Navigation."

FIGURE 1-1
Main Menu for the Orders Application



The menu in Figure 1-1 illustrates the following features:

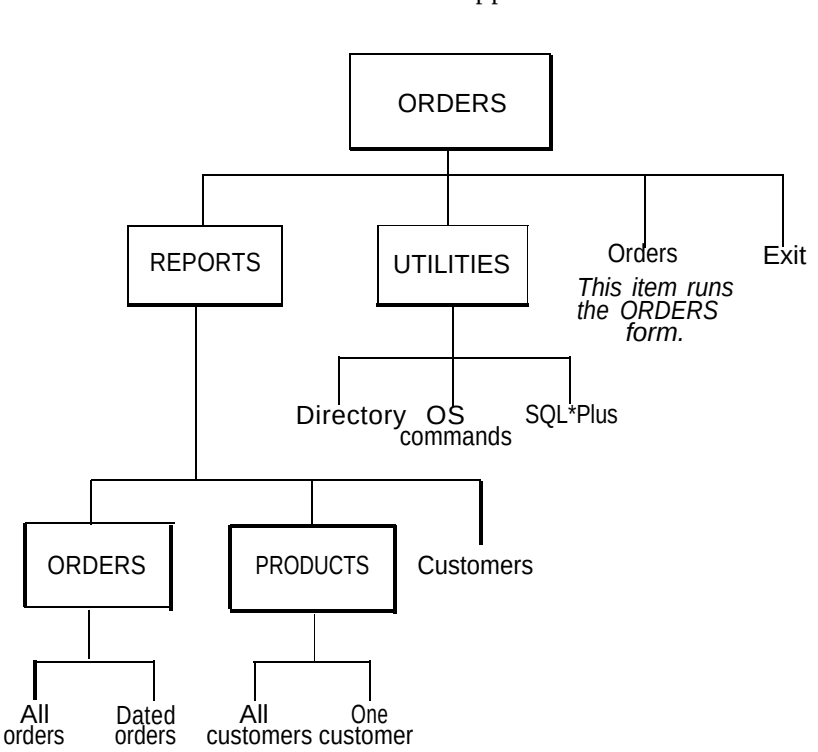
Title bar	In Figure 1-1, the title bar contains the application identification, `Sales Order Application`.
List of items	Each menu has a list of one or more menu items. In Figure 1-1, the items are identified by their short names: `Reports`, `Utilities`, `Orders`, and `Exit`.
Highlighted current choice	In Figure 1-1, the first menu item, `Reports`, is the current menu item. When a menu first appears, the first menu item is always the current choice.
Text for current choice	In Figure 1-1, the full item text for the `Reports` choice appears in the message line near the bottom Of the screen: `Report on sales orders and customers`.

Application name	The name of the application appears on the status line, which is the bottom line of the screen. In Figure 1-1, `ORDERS` is the application name.
Options available	In the status line, `<BGM>< <OSC>,` and `<DBG>` indicate that the current user has access to a background menu, operating system commands, and debug mode, respectively.
Editing mode	In the status line, `<Rep>` indicates replacement mode (typed characters overwrite existing characters) or `<Ins>` indicates insert mode.

Menu Tree for the Sample Application

The diagram in Figure 1-2 is a menu tree showing the relationships among the menus and menu items of the Orders Application. The main menu, ORDERS, has four items. Two of the items call submenus, one item calls a form, and the fourth item exits the application.

FIGURE 1-2
Sample Menu Tree



The main menu is the parent of the REPORTS menu and the UTILITIES menu. In turn, the REPORTS menu is the parent of two submenus, ORDERS and PRODUCTS, each of which runs two types of reports. The REPORTS menu has a third item that runs another type of report; and the UTILITIES menu has three items that execute commands in the operating system.

The main menu appears when you first enter the application from the operating system or SQL*Menu (Design). If SQL*Forms calls the application, a starting menu other than the main menu can be specified as the root menu, in which case you can only access that menu and its submenus.

A menu tree with a parent/child relationship is called a *nested structure.* To avoid circular menu structures, items in a parent menu should not call menus at a higher level in the menu tree.

Requirements for Using SQL*Menu

To run SQL*Menu, you need a licensed copy of the SQL*Menu software, related software products, and the computer hardware used to run the programs. You also need to learn how to run the software, and you must be authorized to use appropriate parts of SQL*Menu and related software products.

Hardware

Because SQL*Menu can run on many different kinds of computers, it is beyond the scope of this manual to discuss hardware-related issues. For information about running SQL*Menu on your computer, refer to the *ORACLE Installation and User's Guide* for your operating system.

Software

SQL*Menu Version 5.0 requires ORACLE RDBMS Version 6.0. SQL*Menu Version 5.0 can be linked with SQL*Forms Version 3.0 and SQL*Plus Version 3.0.

To use the sample applications discussed in this manual, you must have those applications installed as well.

Related Products

SQL*Menu Version 5.0 uses PL/SQL Version 1.0, Oracle*Terminal Version 1.0, and Oracle Toolkit Version 1.0. It can also run with SQL*ReportWriter Version 1.0 and other Oracle software tools. For information about these Oracle products, see the documentation for each product.

Basic Terminology

To use SQL*Menu, you will need a working knowledge of the terms listed in the Glossary. The Glossary includes terms specific to SQL*Menu and terms pertaining to SQL*Forms and the ORACLE RDBMS.

For detailed explanations of other technical terms, see the *SQL*Forms Designer's Reference* and the *SQL Language Reference Manual.*

Authorization to Use SQL*Menu

To use SQL*Menu, you need the proper authorization:

System username and password If your computer is shared by several people, you need a valid username and password to gain admittance to your operating system.

ORACLE username and password Even if your computer is not shared by others, you must have an ORACLE username and password.

SQL*Menu (Run Menu) enrollment To run menu applications, your ORACLE username must be enrolled in SQL*Menu with operator" privileges. You must also be assigned to a role with access privileges for the SQL*Menu application(s) you need to run.

Note: To run the basic SQL*Menu exercises in this book, you can log in as SCOTT with the password TIGER. This username has already been granted permission to use every menu item in the sample applications. Of course, SCOTT must also be enrolled as a valid ORACLE user on your system.

SQL*Menu (Design) enrollment To design menus, your ORACLE username must be enrolled in SQL*Menu with design privileges. (To run a menu application, such as one you are designing, you must also be assigned to a role that has access to that application.)

The Database Administrator

If your computer is shared by others, your organization should have a database administrator (DBA) who is responsible for creating and maintaining usernames and passwords. Ask your DBA to give you a username and password if you don't already have them.

We recommend that your organization assign a SQL*Menu DBA to manage SQL*Menu and to assist its users. The SQL*Menu DBA needs to work closely with the ORACLE DBA and operating system administrator. The SQL*Menu DBA need not be the person who installed the ORACLE RDBMS. A SQL*Menu DBA is any user who has the DBA privilege for SQL*Menu.

If you are working on a single-user system, you may be expected to perform the SQL*Menu DBA's functions. (See the *ORACLE RDBMS Database Administrator's Guide* for information on general ORACLE database administration procedures.)

Installing SQL*Menu

This book does not address installing SQL*Menu, because the installation procedures depend on the operating system you are using. For instructions on installing SQL*Menu, see the *ORACLE Insallation and User's Guide* for your operating system.

PART

II

SQL*MENU OPERATOR'S GUIDE

CHAPTER

2

RUNNING A MENU APPLICATION

This chapter explains the basic operations you need to run a SQL*Menu application and illustrates the operations with exercises from the sample Orders Application.

This chapter covers the following topics:

- how to start and exit SQL*Menu (Run Menu)
- how to select menu items
- how to enter a value for a parameter
- how to get help information
- how to use a background menu
- how to use operating system commands

As you read this chapter, you should run the Orders Application and follow the steps outlined in the exercises. The exercises include illustrations of the screen displays in the default (pull-down) display style.

Starting and Exiting SQL*Menu (Run Menu)

The SQL*Menu component that runs menu applications is SQL*Menu (Run Menu). Before you start, you will need the list of SQL*Menu (Run Menu) function key assignments for your keyboard. If necessary, ask your SQL*Menu DBA for help.

For more information about SQL*Menu (Run Menu) function keys, see Chapter 4, "Function Keys for Menu Operators."

Logging In to SQL*Menu (Run Menu)

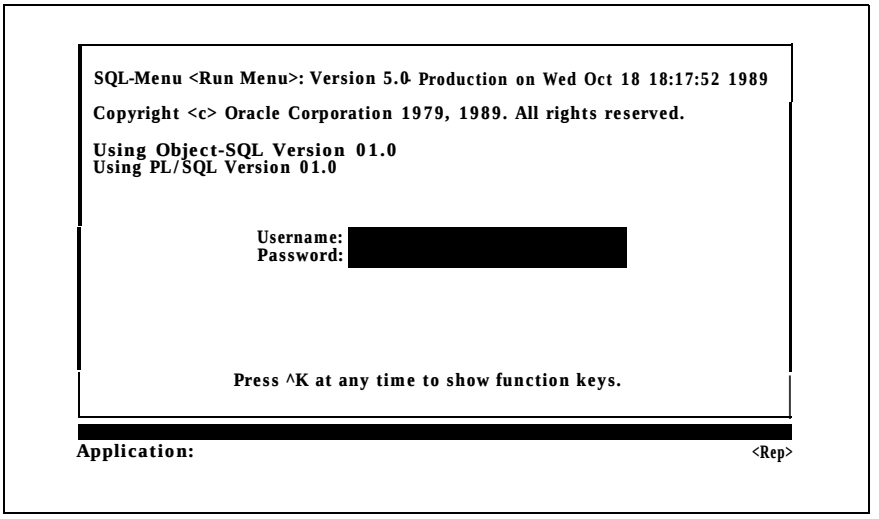
To run the basic SQL*Menu exercises, you can log in as SCOTT (password TIGER). If you want to log in with your own username and password, ask your DBA for authorization.

Exercise 2-1
Starting SQL*Menu

1. Turn on your computer or terminal and logon to your computer's operating system, if required. You should see the operating system prompt on the screen.
2. Type `RUNMENU ORDERS` and press Return. This command starts SQL*Menu running the Orders Application.

   SQL*Menu displays the login screen shown in Figure 2-1.

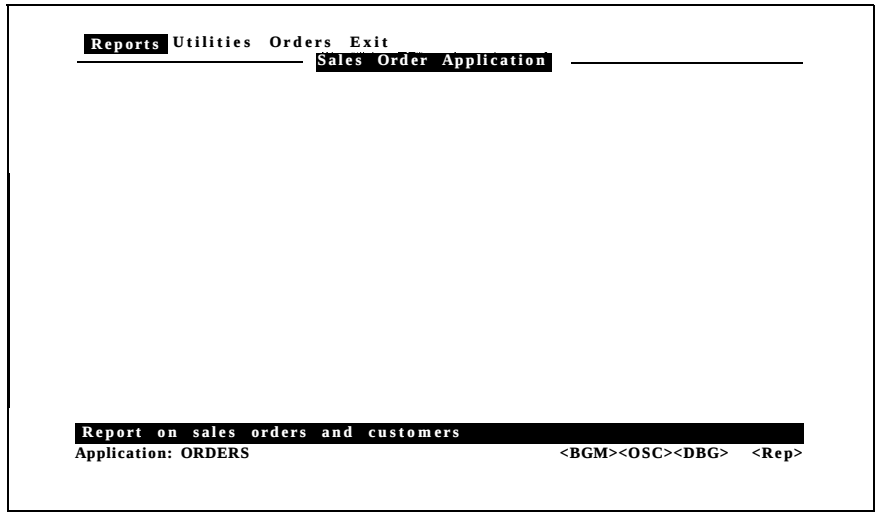
FIGURE 2-1
The SQL*Menu (Run Menu) Login Screen



3. Type SCOTT as your username and press [Next Field].
4. Type TIGER as your password and press [Next Field]. For security reasons, your password does not appear on the screen.

The message `Working . . .` appears while the system confirms that you are authorized to use SQL*Menu and run the Orders Application. Then the ORDERS main menu appears on your screen (Figure 2-2).

FIGURE 2-2
Main Menu of the Orders Application



For more information about logging into SQL*Menu (Run Menu), see "SQL*Menu System Components" in Chapter 12.

Exiting SQL*Menu (Run Menu)

The Orders Application has an item in the main menu that allows you to exit SQL*Menu (Run Menu). To select the Exit item, simply type E when you are in the main menu of the Orders Application.

Other applications might not have an Exit item. You can always exit a menu application by pressing [Quit], or you can go to the application menu and either exit SQL*Menu (Run Menu) or select another application to run.

Using the Application Menu

The application menu allows you to exit SQL*Menu (Run Menu) or to start running another application. To leave a menu application and log out of SQL*Menu (Run Menu) by way of the application menu, first press [Application Menu].

SQL*Menu (Run Menu) displays the application menu, which lists all the applications you are authorized to run. The final item on the application menu is always the Exit item (Figure 2-3). You can exit by selecting this item or by pressing [Cancel].

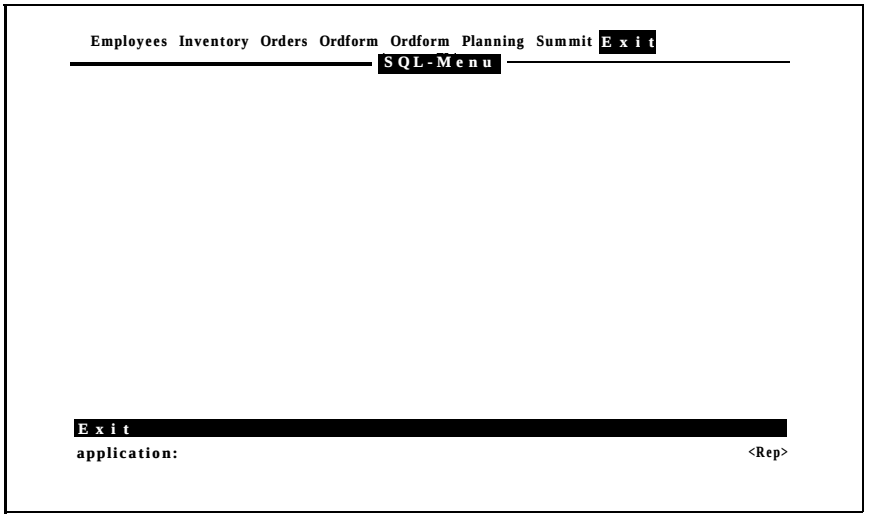
If the application menu contains many items, the Exit item might not appear on the screen. You can move the cursor to off-screen items by pressing [Right] or [Left].

When the cursor is on the first item in the application menu, you can move to the Exit item by pressing [Left] once. Move the cursor to the Exit item, then select it by pressing [Select] or Return.

Note: If you are using the full-screen menu display style, press [Down] and [Up] instead of [Right] and [Left], respectively.

Figure 2-3 shows the Exit item in an application menu.

FIGURE 2-3
The Exit Item in the Application Menu



Selecting Menu Items

When a pull-down menu first appears, the current menu item is the leftmost item, which is highlighted to show the cursor position. You can select the current item by pressing [Select] or Return.

To select an item other than the current one, you can either enter the first capital letter of the item you want, if that capital letter is unique in the menu, or move the cursor to that item and press [Select] or Return.

In a bit-mapped environment, you can also select a menu item by clicking on it with the mouse.

In a block-mode environment, the menu items are fields you can move to by pressing Tab. Pressing [Select] or Enter transmits your selection to the host computer. SQL*Menu ignores anything you type in the fields in pull-down and bar style menus. (Menu items are not highlighted in a block-mode environment.)

The exercises that follow show you how to use these two methods of selecting menu items in the sample Orders Application. The menus are displayed in the standard pull-down menu style.

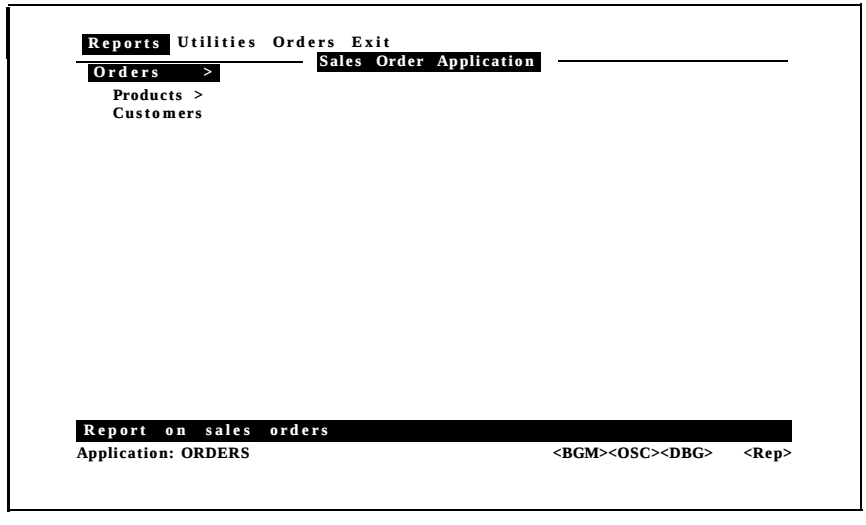
For information about selecting menu items in full-screen and bar style menus, see Chapter 3, "Menu Styles and Navigation."

Exercise 2-2
Selecting the Current Item

1. Start the Orders Application by following the steps of Exercise 2-1. The main menu appears with the cursor on the Reports item (Figure 2-2).

2. Press [Select] to select the highlighted Reports item. The Reports submenu appears, as shown in Figure 2-4. (The character > to the right of an item name indicates that the item calls a submenu.)

FIGURE 2-4
Reports Submenu of the Orders Application



Selecting by Letter

To select a menu item by letter in a pull-down menu, you type the first capital letter of the item name. You do not have to press a function key to activate the item.

The first capital letter does not have to be the first letter of the name. For example, if the Exit item were spelled eXit, you would type X to activate it.

If more than one item in the menu has the same first capital letter, the selection depends on cursor position from the currently highlighted item, SQL*Menu selects the next item to the right (or below, in a pull-down submenu) having that first capital letter.

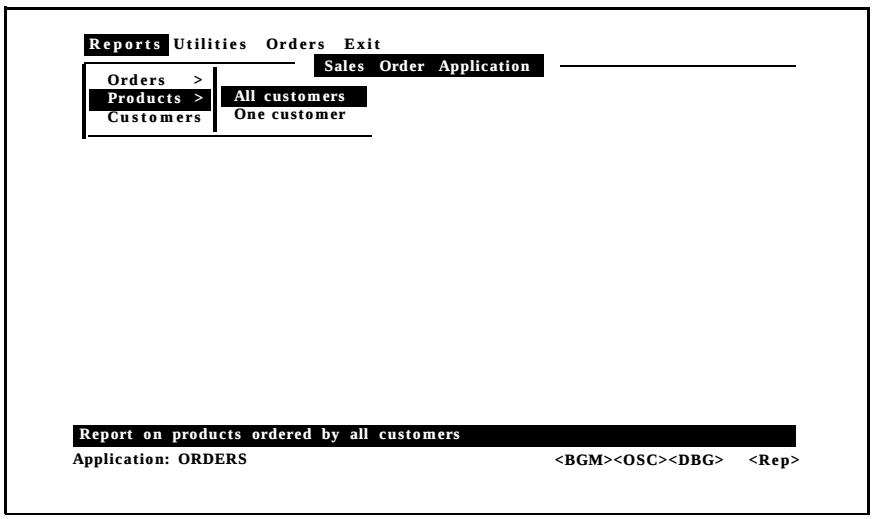
For example, if an application menu contains application names that begin with E, you cannot exit SQL*Menu (Run Menu) by pressing [Application Menu] and immediately typing E because that would select the leftmost application beginning with E. Instead, you would have to move the cursor in the application menu before selecting Exit.

Exercise 2-3 illustrates the letter-entry method of selecting menu items.

Exercise 2-3
Selecting an Item Letter Entry

1. Enter the Reports submenu of the Orders Application as described in the previous exercises.
2. In the Reports submenu, type P to select the Products item. The Product Report submenu appears, as shown in Figure 2-5.

FIGURE 2-5
Product Report Submenu



Selecting by Cursor Movement Keys

To select a menu item by moving the cursor, you use the cursor movement function keys to highlight the item you want, and then press [Select] or Return.

Cursor movement keys perform the following actions:

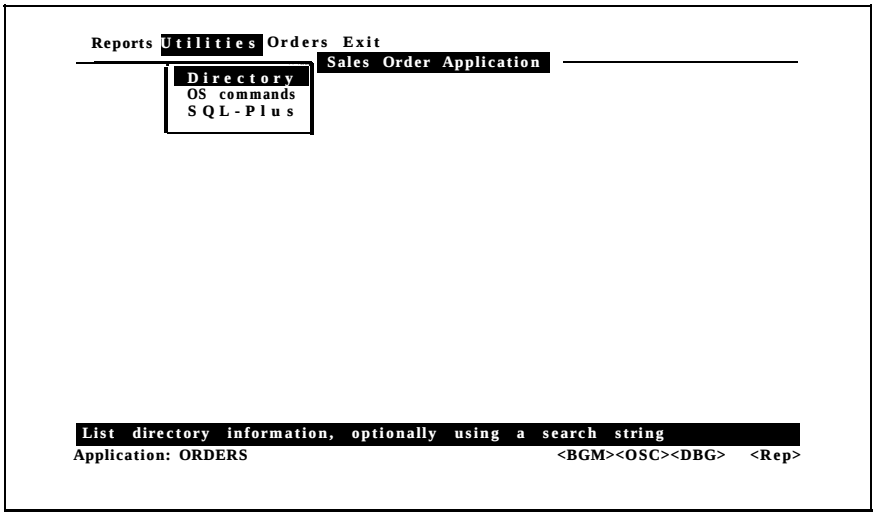
- [Left] and [Right] move to the adjacent item in the main menu (except in the full-screen menu display style).
- [Up] and [Down] move to the adjacent item in a pull-down submenu, or return to the top or bottom item from the bottom or top, respectively.
- In a pull-down submenu of the main menu, [Left] and [Right] move directly between submenus.
- [Main Menu] returns to the main menu from any submenu, in any menu display style.
- In a pull-down submenu, [Cancel] returns to the parent menu.

Exercise 2-4 illustrates the use of cursor movement keys.

Exercise 2-4
Selecting an Item: Cursor Keys

1. In the Product Report submenu, press [Cancel]. The Reports submenu reappears (Figure 2-4).
2. Press [Right] to move to the Utilities submenu, as shown in Figure 2-6.

FIGURE 2-6
Utilities Submenu of Orders Application



3. Press [Right] again. The cursor moves to the Orders item of the main menu (Figure 2-2). This item calls a form, so there is no submenu to display.

4. Press [Right] twice more to move through the Exit item back to the Reports submenu (Figure 2-4).
5. Press [Down] to move to the Products item, then press [Select] to move to the Product Report submenu (Figure 2-5).
6. Press [Cancel] to return to the Reports submenu.
7. Press [Up] to move from the Orders item to the Customers item.
8. Press [Main Menu] to move to the main menu.

For more information about function keys in SQL*Menu (Run Menu), see Chapter 4, "Function Keys for Menu Operators."

Entering Parameter Values

When SQL"Menu requires a value before it can perform the next action, a *dialog box* appears prompting you for the value. For example, you might have to enter a file name or the name of a form to run.

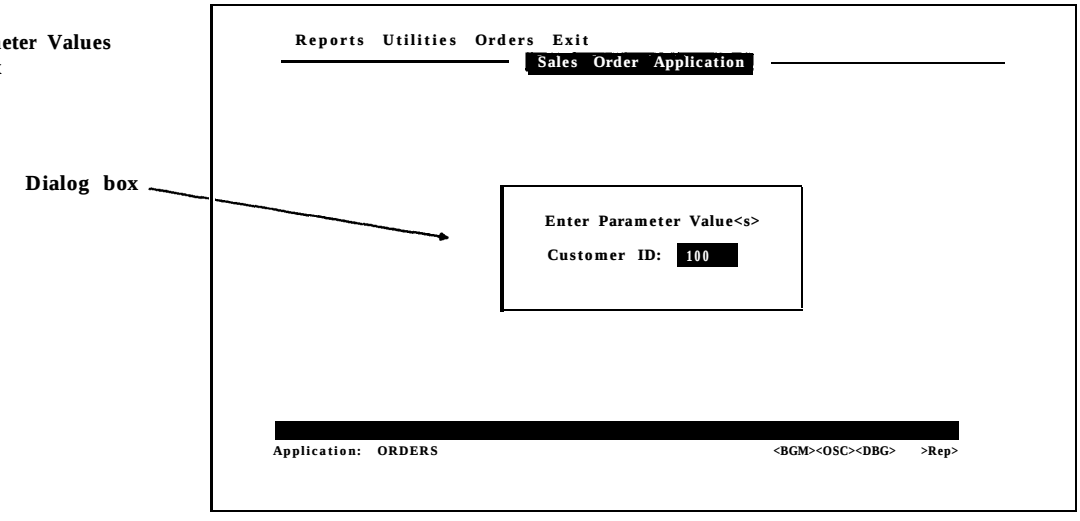
To enter a value in a dialog box, type the appropriate information and press [Accept]. If the dialog box has two or more data-entry fields, press [Next Field] to move the cursor to another field or [Previous Field] to return to a field.

Exercise 2-5 illustrates how to use a dialog box in the sample Orders Application. Because the designer of the Orders Application has already stored command files that generate reports, you can produce a report by selecting the appropriate menu item. To run a report on all products ordered by a particular customer of Summit Sporting Goods, you specify the customer ID in a dialog box.

Exercise 2-5
Running a Report

1. In the main menu of the Orders Application, type R to select the Reports item. The Reports submenu appears (Figure 2-4).
2. Type P to select the Products item. The Product Report submenu appears (Figure 2-5).
3. Type O to select the One Customer item. The Enter Parameter Values dialog box appears with a default value of 100 in the Customer ID field, as shown in Figure 2-7.

FIGURE 2-7
Enter Parameter Values Dialog Box



4. Type 106 and press [Accept] to run a report on the products ordered by customer number 106.

   SQL*Menu calls SQL*Plus, which produces a report from the demonstration tables for the Orders Application, as shown in Figure 2-8.

FIGURE 2-8
Report Products Ordered by Customer 106

SUMMIT SPORTING GOODS - PRODUCTS ORDERED

CustID	Customer	Product	Price	Quantity	Total	Order	Date
106	SHAPE UP	100860	$44.00	10	$440	604	15-JUN-86
		100861	$42.00	2	$84	604	15-JUN-86
			$45.00	100	$4,500	605	14-JUL-86
		100870	$2.80	500	$1,400	605	14-JUL-86
		100890	$58.00	3	$174	604	15-JUN-86
			$58.00	5	$290	605	4-JUL-S6
		101860	$24.00	50	$1,200	605	14-JUL-86
		101863	$9.00	100	$900	605	14-JUL-86
		102130	$3.40	10	$34	605	14-JUL-86
		200376	$2.40	1	$2	601	0l-MAY-86

Press RETURN to return to SQL-Menu . . .

5. Press Return to leave the report and return to the main menu.

Getting Help Information

Several kinds of online help are available to operators of menu applications:

- hints and error messages
- function key help
- application help system (optional)

The following sections describe how to get online help for SQL*Menu (Run Menu). If you need more than the online help, you can lookup the relevant information in this book by using the index or the table of contents.

Context

Most of the online help for SQL*Menu (Run Menu) is *context-sensitive,* that is, the help displayed on your screen refers to the current position of the cursor in the program.

The Message Line

The message line near the bottom of the screen displays hints and error messages that can help you use SQL*Menu (Run Menu). Check the message line first whenever you want help about how to run a menu application.

Hints

The hints and descriptions in the message line are context-sensitive. For example, when the cursor is in the Username field of the login screen, the message line prompts you' to enter your ORACLE username (Figure 2-1). When the cursor is on a menu item, the message line displays a hint for that item.

Error Messages

Error messages appear in the message line to help you diagnose and correct errors. Each error message begins with a prefix and message number, for example, `MNU- 10200: Illegal function in this context` . This error message appears if you press a function key that does not work when the cursor is in its current position.

You can recognize SQL*Menu errors by the prefix `MNU`. See Appendix B to find the probable cause and suggested action for SQL*Menu errors.

Other error messages might be issued by the ORACLE RDBMS (prefix `ORA`), SQL*Forms (prefix `FRM`), or your operating system. For RDBMS errors, see *ORACLE Error Messages and Codes Manual;* for SQL*Forms errors, see the *SQL*Forms Operator's Guide;* and for operating system errors, see your operating system manual.

Function Key Help

To display a list of the function keys available in the current context and their mappings for your terminal, press [Show Keys].

The Application Help System

Context-sensitive help information may be provided by the application designer. If the menu application you are running has this help, you can learn more about a specific menu item by moving the cursor to it and pressing [Help].

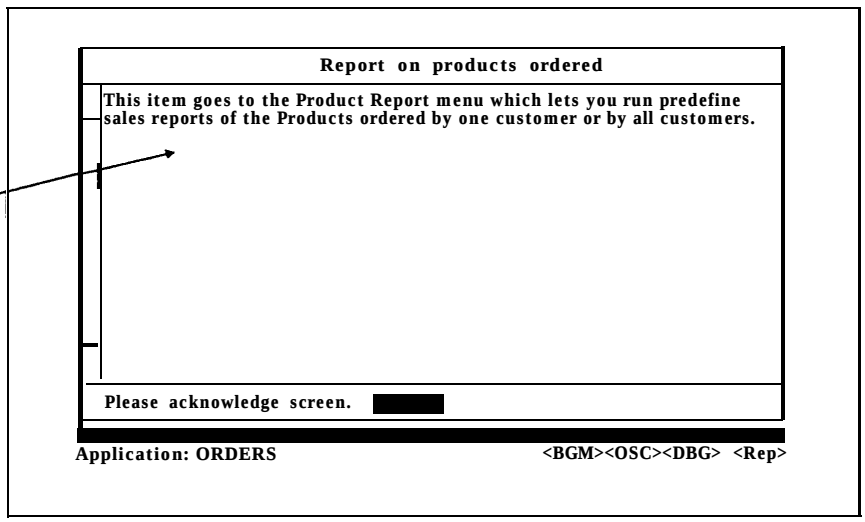
Note: A comprehensive online help system is available in SQL*Menu (Design) for the menu design, maintenance, and security operations described in Part III of this book.

Exercise 2-6 illustrates how to display help information for the Products item in the Reports submenu.

Exercise 2-6
Getting Application Help

1. In the main menu of the Orders Application, type R to select the Reports item. The Reports submenu appears (Figure 2-4).
2. Press [Down] to move to the Products item.
3. Press [Help]. Help information for the current item appears on the screen (Figure 2-9).

FIGURE 2-9
Help for the Products Item



4. Press [Select] to acknowledge the help information. The help information disappears and you return to the Reports submenu.

Using the Background Menu

The *background menu* can provide shortcuts and general functions for running an application. If a menu application has a background menu, the background menu lists the functions assigned to the function keys [Background Menu *n*], where *n* is 1 to 10 for the items numberd 1 to 10 on the background menu.

If the notation `<BGM>` appears in the status line at the bottom of your screen, you can use the background menu and its function keys in the current application.

If `<BGM>` does not appear in the status line, either your role does not have the background menu privilege or a background menu does not exist for the application.

You can display the background menu from any other menu in an application by pressing [Show BGM]. It cannot be displayed when a form is on the screen. The background menu always appears as a modified version of the full-screen menu display style.

You can activate background menu items from any menu by using the function keys [Background Menu *n*], where *n* is 1 to 10. Your DBA can give you a list of the actions represented by each item, or you can just display the background menu to seethe list of items and their key mappings.

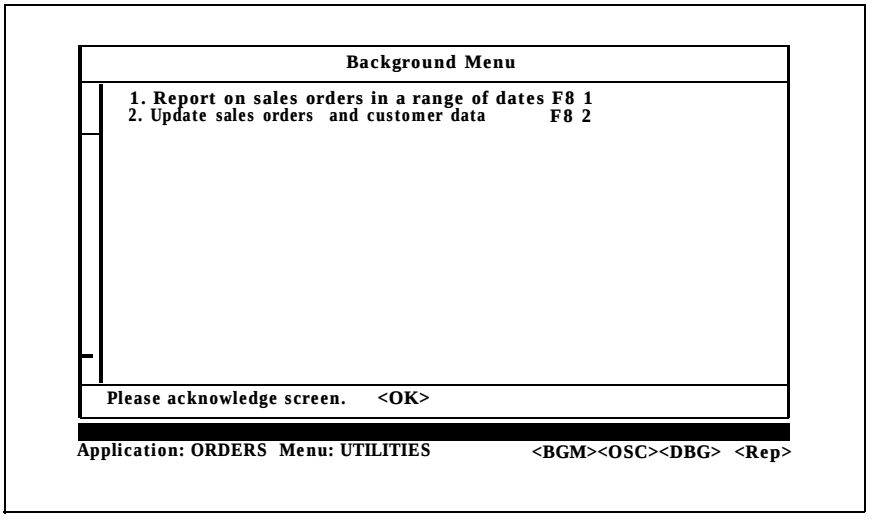
Exercise 2-7 illustrates the use of the background menu for the sample Orders Application.

Exercise 2-7
Using the Background Menu

1. While running the Orders Application, press [Show BGM]. The background menu appears, as shown in Figure 2-10.

   This background menu contains two items, which perform the same actions as other items in the Orders Application. Item 1 is the same as the Dated Orders item in the Order Report submenu, and item 2 is the same as the Orders item in the main menu. The key mappings appear to the right of the text that identifies each item.

FIGURE 2-10
Background Menu for Orders Application



2. Press [Background Menu 2] to select item 2 from the background menu.

   SQL*Menu calls the ORDERS form, which you can use to enter, update, or query information about sales orders for Summit Sporting Goods.

3. Press the SQL*Forms function key [Exit/CancelJ to return to SQL*Menu.

4. Press [Background Menu 1] to select item 1 from the background menu.

   The Enter Parameter Values dialog box appears, prompting you for a starting date and an ending date for the report.

5. Type `01-JuL-86` in the Starting Date field and press [Next Field].

6. Type `31-JUL-86` in the Ending Date field and press [Next Field].

   SQL*Menu calls SQL*Plus and produces a report from the demonstration tables for the Orders Application.

7. Press Return to leave the report and return to SQL*Menu.

Using Operating System Commands

The notation <OSC>, displayed in the status line at the bottom of the screen, shows that you have been granted authority to issue operating system commands from within SQL*Menu. This feature lets you move outside SQL*Menu temporarily to print files, read electronic mail, or perform other tasks not directly available from within the current menu.

You can use one of two function keys to enter commands. Each makes the operating system prompt appear.

- Press [Enter 1 OS Command] to enter a single operating system command. When the command has been carried out, you are prompted to press a key to return to SQL*Menu.
- Press [Enter >1 OS Command] to enter more than one operating system command. To return to SQL*Menu, you must press Return on a blank line.

The Orders Application also provides a Utilities submenu that provides access to the operating system. The OS Commands item in the Utilities submenu performs the same function as [Enter >1 OS Command].

Exercise 2-8
Using an Operating System Command

1. In the Orders Application, press [Enter >1 OS Command].

   The operating system prompt appears.

2. Enter the operating system command that displays the contents of the current file directory on your computer. For example, on some systems you type `ls` or `dir`:

   ```
   OS>dir  [Return]
   ```

   The operating system lists the files in the current directory, followed by the message `Type any key to continue. . . .`

3. Press Return on a blank line to return to SQL*Menu.

CHAPTER

3

MENU STYLES AND NAVIGATION

This chapter describes the menu display styles that SQL*Menu provides for running menu applications, and it explains how to navigate in each menu display style. Three different display styles are available

- pull-down menus (Oracle standard menus)
- bar menus
- full-screen menus

The default style is the standard pull-down menu display.

Navigating in SQL*Menu (Run Menu)

You can go from one menu to another by using menu items or function keys. In the full-screen display style, you can also enter the name of a menu or menu item in the current application to navigate directly to that menu or item.

To navigate through the menu tree by standard paths, you select the menu items that call submenus. The following function keys are available for navigation in SQL*Menu (Run Menu):

- [Right] and [Left] move the cursor from a menu item to the adjacent menu item in a bar style menu or in the main menu of an application run in the pull-down style.

  In a pull-down style submenu of the main menu, [Right] and [Left] move the cursor to the submenu called by the adjacent item in the main menu or to the adjacent item in the main menu if that item does not call a submenu.

- [Up] and [Down] move the cursor to the adjacent menu item in a pull-down style submenu.
- [Next Field] and [Previous Field] move the cursor to the adjacent menu item in a bar or pull-down menu display style, or from a field to the adjacent field in a form or dialog box.
- [Previous Menu] moves the cursor to the previous menu from a submenu. From the main menu, [Previous Menu] moves the cursor to the application menu.
- [Main Menu] moves the cursor to the main menu of the current application.
- [Show BGM] moves the cursor to the background menu, if one is available for the current application and if you have the background menu privilege (indicated by `<BGM>` in the status line).
- [Application Menu] moves the cursor to the application menu, in which you can select an application to run or exit SQL*Menu (Run Menu).
- In a pull-down style menu, [Cancel] moves the cursor from a submenu to its parent menu or exits the application from the main menu. In a bar or full-screen style menu, [Cancel] exits the current application.
- [Quit] exits SQL*Menu (Run Menu).

For more information about SQL*Menu (Run Menu) function keys, see Chapter 4, “Function Keys for Menu Operators.”

Specifying a Menu Display Style

All menu styles display the same list of menu items, but the text that identifies a menu item in the full-screen display style differs from the short name used in pull-down and bar menus.

You can specify a display style when you log in to SQL*Menu, or you can use the default style specified for your system. This is the syntax for specifying a menu style when you log in to SQL*Menu (Run Menu):

```
RUNMENU -M {PULLDOWN |BAR |FULLSCREEN}
```

or:

```
RUNMENU -M {P |B |F}
```

For more information about the syntax of the RUNMENU command, see "SQL*Menu System Components" in Chapter 12.

Using Pull-Down and Bar Style Menus

The primary difference between the pull-down style and the bar style is the representation of submenus:

- Applications running in bar style show all submenus in a horizontal menu bar at the top of the screen.
- Applications running in pull-down style keep the main menu on display in the menu bar and show submenus below their corresponding items in the main menu.

Menu items are identified by their short names, which can be single words or multiple words (separated by blanks). Item names in the menu bar are separated by two spaces, and item names in pull-down submenus are displayed in a vertical list.

If the items cannot all be displayed on the screen, you can scroll to off-screen items by using the cursor movement keys. Items that are not available to you either do not appear in the menu or appear without bold characters. The cursor cannot move to an unavailable item.

In both pull-down and bar display styles, the message line displays a hint for the current (highlighted) item.

To select a menu item in an application running in the pull-down menu style, use either of the two methods described in Chapter 2:

- Highlight the item by moving the cursor, then press [Select].
- Type the first capital letter of the item name. (If the first capital letter is not unique, the selection depends on cursor position.)

Pull-Down Style Menus

The pull-down menu display style is the Oracle standard style. The exercises in Chapter 2 demonstrate how to run the Orders Application in the pull-down display style.

Figures 3-1 and Figure 3-2 show first-level and second-level submenus from the Orders Application.

FIGURE 3-1
First-Level
Pull-Down Submenu

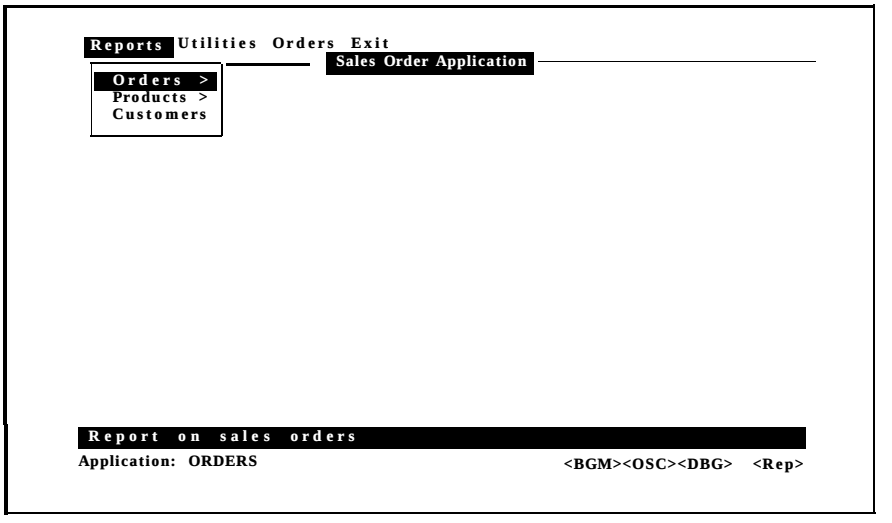
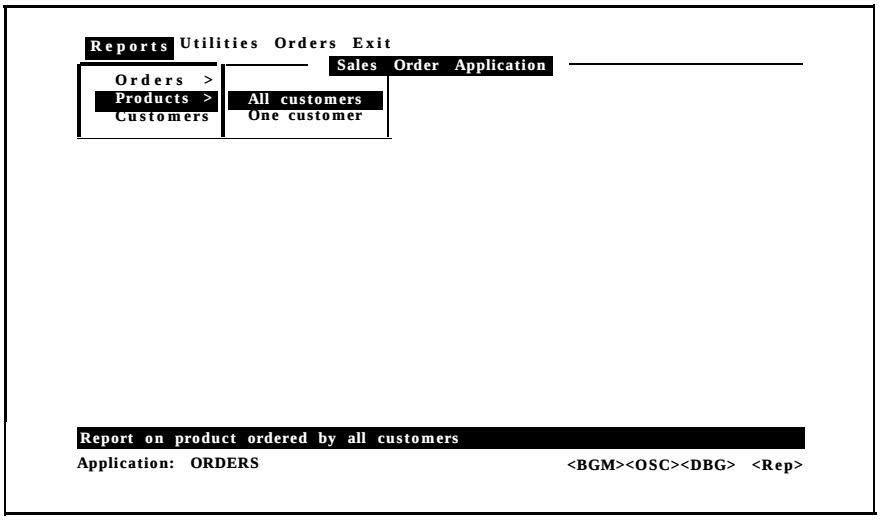


FIGURE 3-2
Second-Level
Pull-Down Submenu



Bar Style Menus

When you select a menu item that calls a submenu in the bar display style, the new submenu items appear in the menu bar, replacing the previous menu display.

Figures 3-3 and 3-4 show the bar-style submenus that correspond to Figures 3-1 and 3-2, respectively.

FIGURE 3-3
First-Level
Bar Style Submenu

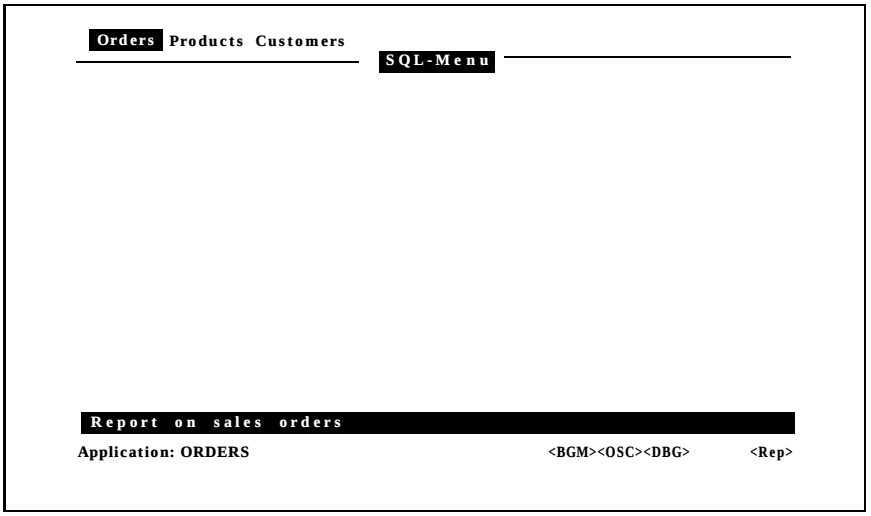
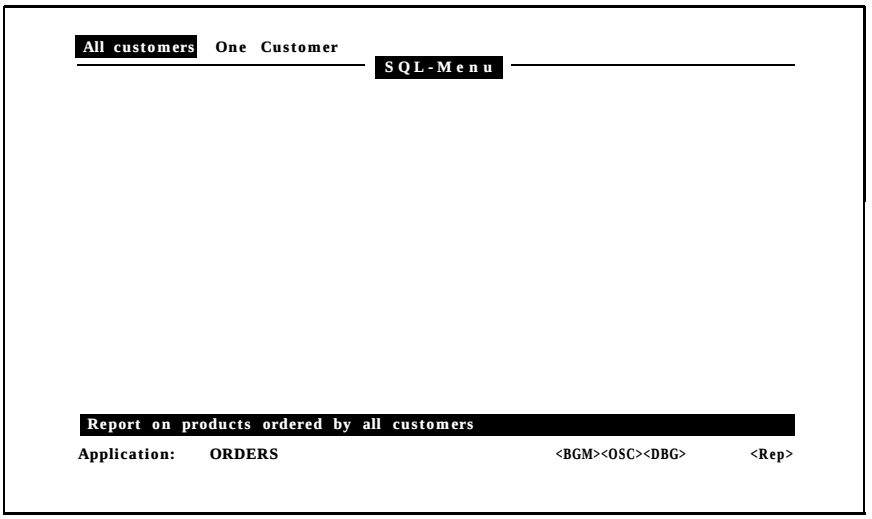


FIGURE 3-4
Second-Level
Bar Style Submenu



Using Full-Screen Menus

A full-screen style menu displays menu items in a vertical numbered list, with the identifying text that would appear as a hint for a bar or pull-down style menu. Highlighting indicates the current item, and the Enter Your Choice field contains the item number.

Figures 3-5 and 3-6 show the full-screen submenus that correspond to Figures 3-1 and 3-2, respectively.

FIGURE 3-5
First-Level
Full-Screen Submenu

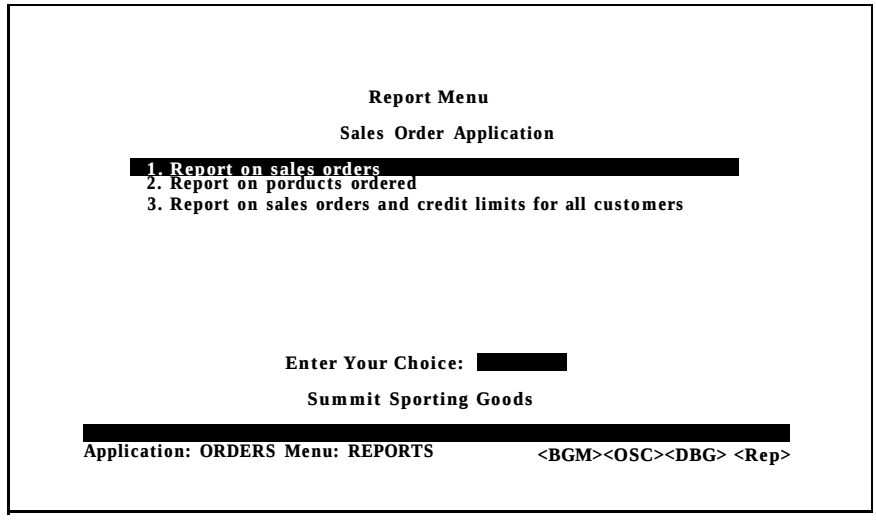
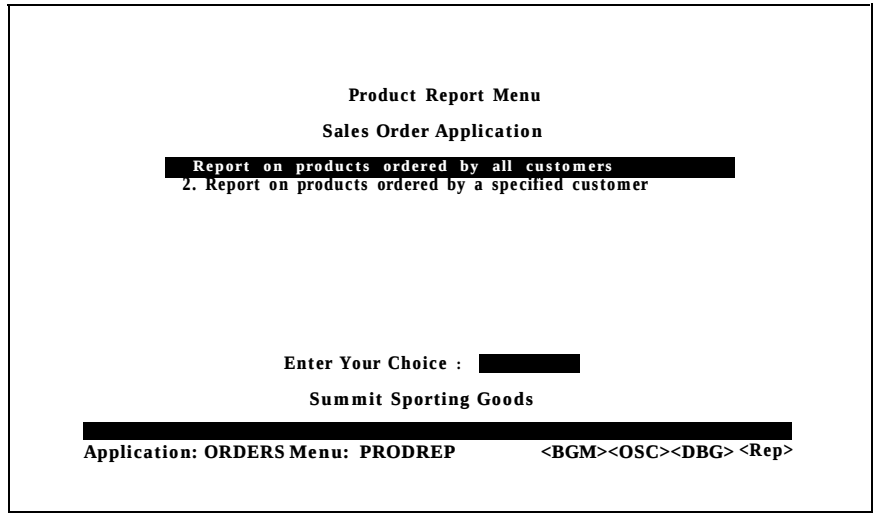


FIGURE 3-6
Second-Level
Full-Screen Submenu



Navigating and Selecting Items in Full-Screen Menus

You can use the following methods to selector navigate to a menu item in an application running in the full-screen display style:

- Select any item in the current menu by typing its item number in the Enter Your Choice field and pressing [Select] or Return.
- Move the highlighting to an item in the current menu by pressing [Down] or [Up], then select it by pressing [Select) or Return.

  Note: [Down] and [Up] can move the highlighting to items that do not appear on the screen, if the menu contains more items than one screen can display.

- Navigate to a menu or menu item by direct selection, as described in the following section.

Navigating by Direct Selection

In the full-screen menu display style; you can move to a menu or select a menu item directly from any location in the menu tree..

- Navigate directly to a menu by typing the menu name in the Enter Your Choice field and pressing [Select] or Return.
- Select a menu item directly by typing the menu name, a hyphen (-), and the item number in the Enter Your Choice field and pressing [Select] or Return.

To learn the names of menus, you can look at the status line or use the Where option, which is explained later in this chapter.

The status line shows the name of the current menu. The Where option displays the names of the current menu and its parent menu, if it has one, and all parent menus above them on the same branch of the menu tree.

As you operate an application in the full-screen display style, you will become familiar with the menu names and their item numbers so that you can navigate easily.

Exercise 3-1 demonstrates how to navigate and select items directly in the sample Orders Application.

Exercise 3-1
Selecting Menus and Items Directly

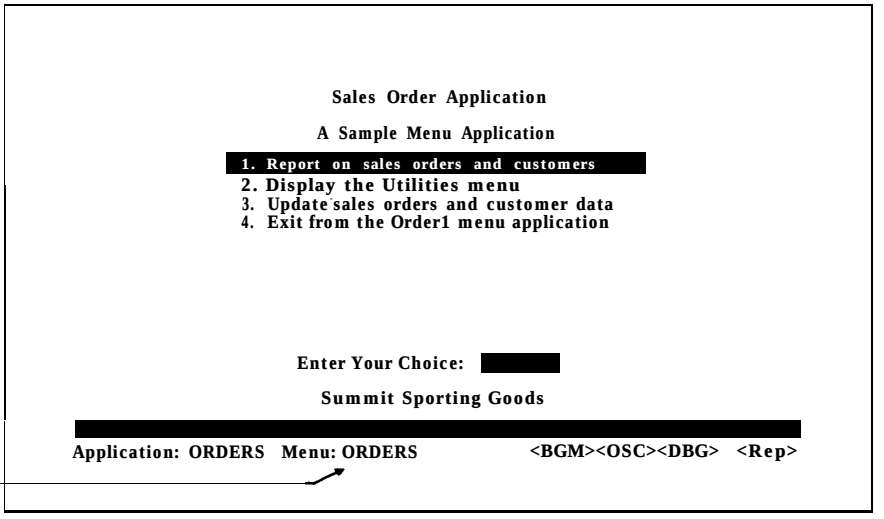
1. At the operating system prompt, invoke the Orders Application in the full-screen display style by entering the following command:

   ```
   RUNMENU ORDERS -M FULLSCREEN
   ```

2. In the login screen, enter the username SCOTT and password TIGER.

   The main menu of the Orders Application appears in the full-screen style, as shown in Figure 3-7.

FIGURE 3-7
Main Menu in Full-Screen Display Style



3. In the Enter Your Choice field, enter the menu name `PRODREP` to navigate directly to the Product Report menu shown in Figure 3-6.
4. From the Product Report menu, enter `ORFERS-3` to select the third item from the main menu ("Update sales orders and customer data"). This item is the same as the item labeled Orders in pull-down and bar style displays.

   SQL*Menu calls the ORDERS form, which you can use to enter, update, and query information about sales orders for Summit Sporting Goods.
5. Press the SQL*Forms [Exit/Cancel] key to return to the main menu of the Orders Application.

Restrictions on Direct Selection of Menus and Items

The following restrictions apply to the use of direct selection and navigation methods in applications running in the full-screen menu display style:

- You cannot use the direct selection and navigation methods to move from the application menu to menus or items in another application.

  For example, when the application menu is displayed you must enter the number or cursor position corresponding to the desired application. You cannot type `ORDERS` to move to the Orders Application.

- Once you are in an application, you cannot use the direct selection method to refer back to the application menu. Instead, you must press [Application Menu].
- Within one application, you cannot refer to menus or items in a different application.

  For example, when you are in the Orders Application you cannot type `INVENT-2` to select the second item in the main menu of an inventory application named INVENT.. Instead, you must press [Application Menu], select the INVENT application, and then navigate to the item.

The Where Option Menu Names and Locations

The Where option can be turned on or off by pressing [Where Display] in an application running in the full-screen menu style. The Where option is not available in pull-down and bar style menus.

When the Where option is turned on in full-screen display mode, the name of the current menu is shown in a horizontal bar across the screen.

The names of any parent menus in the same branch of the menu tree as the current menu are shown to the left of the current menu name. These names form a *menu path.* A menu path helps you find your way in a complex menu structure.

Exercise 3-2 demonstrates how to use the Where option in the Orders Application.

Exercise 3-2 Using the Where Option

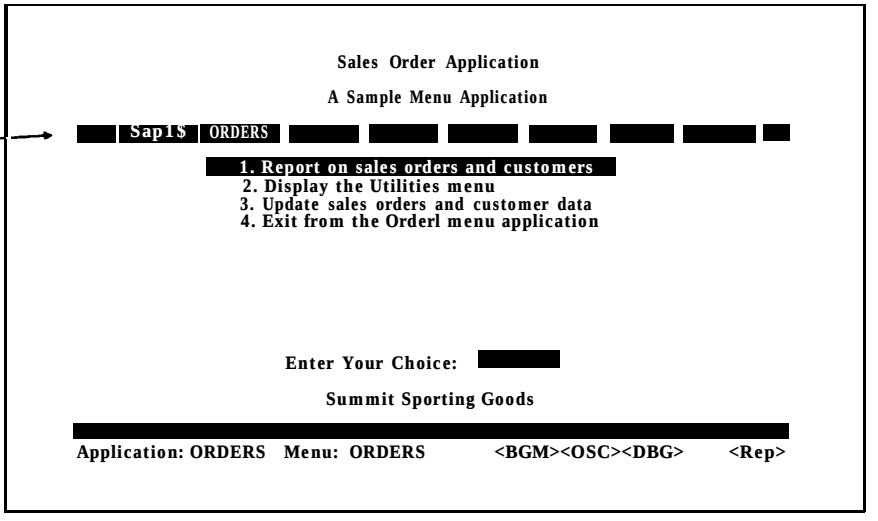
1. At the operating system prompt, enter the following command:

   ```
   RUNMENU ORDERS -M F
   ```

2. In the login screen, enter the username SCOTT and password TIGER.
3. Press [Where Display].

The menu path appears, as shown in Figure 3-8. The symbol `$apl$` on the left represents the application menu.

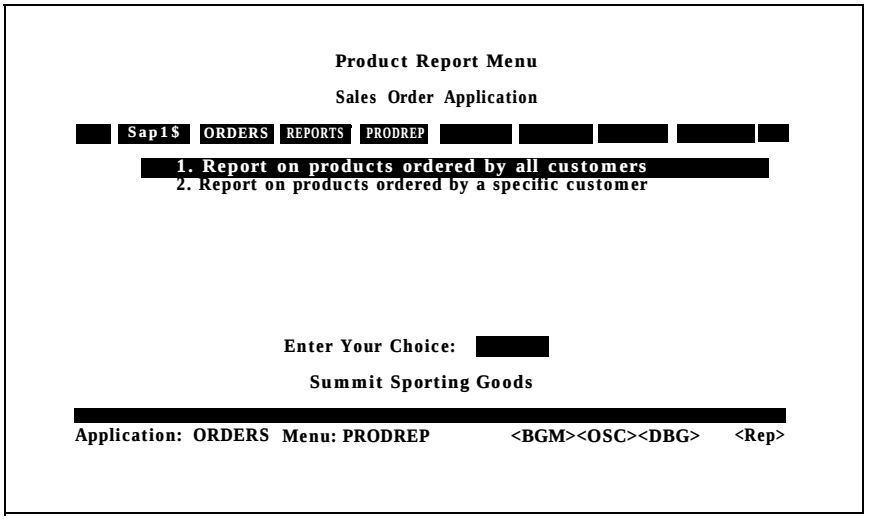
FIGURE 3-8
Where Display The Orders Main Menu



4. Enter `REPORTS-2` to navigate to the Product Report submenu.

   The new menu path appears, as shown in Figure 3-9.

FIGURE 3-9
Where Display Product Report Menu



5. Remove the menu path from the screen by pressing [Where Display] once more.

CHAPTER

4

FUNCTION KEYS FOR MENU OPERATORS

This chapter describes the function keys that you can use in SQL*Menu (Run Menu) and groups them into categories for the operator's interface. The function key groupings include the following categories:

- cursor movement functions
- editing functions
- general functions
- user assistance functions

Function Keys for SQL*Menu (Run Menu)

The SQL*Menu (Run Menu) functions can move the cursor around a screen, navigate to another screen, or execute an action. You perform these functions by pressing function keys. (In a bit-mapped environment, you can perform some functions with either the mouse or a function key.)

This book refers to function keys by their function names, rather than physical key names, because the physical keys that correspond to the functions are different for different computer systems. Function names always appear in square brackets, such as [Left] and [Right].

Many function keys for SQL*Menu (Run Menu) are the same as function keys for SQL*Forms (Run Form), but some functions are unique to either SQL*Menu or SQL*Forms. For information about SQL*Forms (Run Form) function keys, see the *SQL*Forms Operator's Guide.*

Function Key Mappings

A *keyboard map* shows what physical keys correspond to the SQL*Menu functions. SQL*Menu has two keyboard maps, one for SQL*Menu (Run Menu) and one for SQL*Menu (Design).

While you are using SQL*Menu, you can display your terminal's keyboard map by pressing [Show Keys]. The resulting display lists the functions and the physicaI key names for the current context of SQL*Menu (Run Menu).

The login screen shows you what key or combination of keys to press for [Show Keys] on your keyboard. [Show Keys] works within any context of the SQL*Menu (Run Menu) interface, including dialog boxes, forms, and online help.

Function Key Descriptions

This section describes all of the functions available to operators in SQL*Menu (Run Menu). The function keys and their descriptions are listed alphabetically, and the following section groups them by related functions.

Note: The function key descriptions in this section are based on the default function key assignments. Some of these functions may not be available to you. Ask your SQL*Menu DBA for a list of the available functions.

The following function keys are available in SQL*Menu (Run Menu):

[Accept]*	[Enter 1 OS Command]	[Print]*
[Application Menu]	[Enter >1 OS Command]	[Quit]*
[Background Menu *n]*	[Enter Application Parameters]	[Redefine Usr/Pwd]
[Beginning of Line]*	[Enter Menu Parameters]	[Refresh]*
[Cancel]*	[Help]*	[Right]*
[Clear Field]*	[Insert/Replace]*	[Select]*
[Debug Mode]	[Left]*	[Show BGM]
[Delete Backwards]*	[Main Menu]	[Show Keys]*
[Delete Character]*	[Next Field]*	[Up]*
[Down]*	[Previous Field]*	[Where Display]
[End of Line]*	[Previous Menu]	

* These function keys are also used in SQL*Menu (Design).

The specific function that a function key performs can vary depending on the context in which you use it. The descriptions in this section provide context-sensitive information where applicable.

[Accept]

[Accept] terminates data input. It has no function in a menu.

In a dialog box [Accept] closes the dialog box and acts upon your entry.

[Application Menu]

[Application Menu] takes you to the Application Menu, where you can select another application to run.

[Background Menu *n]*

[Background Menu *n*] selects background menu item *n*, where *n* is 1 through 10.

[Beginning of Line]

[Beginning of Line] has no function in a menu.

In a field [Beginning of Line] moves the cursor to the first column (displayed or undisplayed) of the current field.

[Cancel]	[Cancel] terminates the current operation and exits the current context," discarding any changes or selections. **In a pull-down menu** [Cancel] redisplays the previous menu if the current menu is a submenu. From the main menu, [Cancel] exits the menu application. **In a bar or full-screen menu** [Cancel] exits the menu application. **In a dialog box** [Cancel] closes the dialog box and returns the cursor to the main menu.
[Clear Field]	[Clear Field] clears (deletes) the contents of the current field.
[Debug Mode]	[Debug Model enables or disables the debug option within SQL*Menu.
[Delete Backwards]	[Delete Backwards] deletes the character to the left of the current cursor position and moves any characters on the right one space to the left.
[Delete Character]	[Delete Character] deletes the character at the current cursor position and moves any characters on the right one space to the left.
[Down]	[Down] moves the cursor to the next item in a pull-down style menu. After the cursor reaches the last item, it returns to the first one.
[End of Line]	**In a field** [End of Line] moves the cursor to the right of the last character in the current field, scrolling if necessary.
[Enter 10S Command]	[Enter 1 OS Command] temporarily leaves SQL*Menu (Run Menu) to let you enter a single operating system command.
[Enter >1 OS Command]	[Enter >1 OS Command] temporarily leaves SQL*Menu (Run Menu) to let you enter operating system commands. Entering a blank command returns you to SQL*Menu (Run Menu).
[Enter Application Parameters]	[Enter Application Parameters] displays the Enter Parameter Values dialog box, where you can set all of the application's substitution parameters.

[Enter Menu Parameters]

[Enter Menu Parameters]:

- only works in a full-screen menu
- displays the Enter Parameter Values dialog box, where you can set all of the current menu's substitution parameters

[Help]

[Help] displays any existing help text for the current context.

[Insert/Replace]

[Insert/Replace] switches data entry mode between Insert and Replace:

- In Replace mode, newly typed characters replace any existing characters, beginning at the current cursor position.
- In Insert mode, newly typed characters are inserted at the current cursor position, pushing the character already at that position (and any characters after it) to the right.

[Insert/Replace] functions in any data-entry context. The status line shows the current data-entry mode with `<Ins>` or `<Rep>`.

[Left]

In a menu [Left]:

- highlights the menu item to the left of the current choice in a bar style menu or the main menu of a pull-down style menu application
- moves the highlight from the leftmost item to the rightmost menu item

In a field [Left] moves the cursor one character position to the left.

[Main Menu]

[Main Menu] returns you to the main menu.

[Next Field]

In a menu [Next Field]:

- moves the cursor down or to the right, to the next item in a menu
- moves the cursor back to the first menu item from the last one

In a form or dialog box [Next Field]:

- moves the cursor in sequence, to the next field
- moves the cursor from the last field back to the first field

[Previous Field]

In a menu [Previous Field]:

- moves the cursor up or to the left to the previous item in a menu
- moves the cursor from the first menu item to the last one

In a form or dialog box [Previous Field]:

- moves the cursor in sequence to the previous field
- moves the cursor from the first field to the last field

[Previous Menu]

[Previous Menu] returns you to the previous menu if you are in a submenu. If you are in the main menu, [Previous Menu] returns you to the Application Menu.

[Print]

[Print] displays the Print dialog box, which allows you to create an output file for the current screen.

[Quit]

[Quit] exits SQL*Menu (Run Menu) and returns to the operating system or to SQL*Menu (Design), if it invoked SQL*Menu (Run Menu).

[Redefine Usr/Pwd]

[Redefine Usr/Pwd] redisplays the login screen so you can log in under a different username, without exiting from SQL*Menu. It automatically logs out your original username. After you enter a username and password, SQL*Menu returns to the application menu.

[Refresh]

[Refresh] restores the screen image after it has been changed by some external cause (for example, by an operator message). [Refresh]:

- refreshes the current screen
- redisplays any changes you have made to the screen if you are on a character mode terminal
- discards any changes you have made to the screen and redisplays previous values if you are on a block mode terminal. (This is a terminal function, not a SQL*Menu function, on block mode devices.)

[Right]

In a menu [Right]:

- highlights the menu item to the right of the current selection
- moves the highlight from the rightmost menu item to the leftmost item

In a field [Right] moves the cursor one character position to the right.

[Select]

In a menu [Select] chooses the indicated menu item.

[Show BGM]

[Show BGM] displays the background menu, if one is available for the current menu application.

[Show Keys]

[Show Keys] displays the keyboard map that corresponds to the terminal definition you specified (or to which the system defaulted) when you logged in to SQL*Menu.

Note: [Show Keys] only displays mappings for the keys that are available in the current context.

[Up]

[Up] moves the cursor to the previous item in a pull-down menu. After the cursor reaches the first item, it skips to the last one.

[Where Display]

[Where Display] in a full-screen style menu displays the menu path, which is the list of menu names that lead to the current menu.

Functional Groupings for Operators

This section places the SQL*Menu (Run Menu) function keys into groups by their basic characteristics:

- cursor movement functions
- editing functions
- general functions
- user assistance functions

Use these groupings to help locate a function when you know what you want to do, such as moving the cursor or deleting data. Once you find the name of the function key for which you want information, you can find its description in the alphabetical listing and find further information by using this book's index.

Note: Some keys appear in more than one group.

Cursor Movement Functions

[Application Menu]
[Beginning of Line]
[Cancel]
[Down]
[End of Line]
[Left]
[Main Menu]
[Next Field]
[Previous Field]
[Previous Menu]
[Right]
[Show BGM]
[Up]

Editing Functions

[Beginning of Line]
[Clear Field]
[Delete Backwards]
[Delete Character]
[End of Line]
[Insert/Replace]
[Left]
[Right]

General Functions

[Accept]
[Cancel]
[Quit]
[select]

User Assistance Functions

[Background Menu *n*]
[Debug Mode]
[Enter 1 OS Command]
[Enter >1 OS Command]
[Enter Application Parameters]
[Enter Menu Parameters]
[Help]
[Print]
[Redefine Usr/Pwd]
[Refresh]
[Show BGM]
[Show Keys]
[Where Display]

PART

III

SQL*MENU DESIGNERS GUIDE

CHAPTER

5

DESIGN OPERATIONS

This chapter explains how to perform design operations and administrative operations in SQL*Menu (Design). These operations are presented in functional groups:

- starting and quitting SQL*Menu (Design)
- basic design operations
- setting options in SQL*Menu (Design)
- application maintenance operations
- security operations

The interface elements that you use in SQL*Menu (Design) include forms, spread tables, dialog boxes, and lists of values. These elements contain data-entry elements such as fields, check boxes, scroll regions, and buttons. As this chapter describes operations in the design interface, it briefly describes how to use each element.

For complete information about interface elements, see Chapter 6, "Elements in the Design Interface."

Overview of Menu Design

A SQL*Menu application consists of *objects* that you define in the SQL*Menu (Design) interface. These objects include the application itself, the menus, the menu items, and any substitution parameters and PL/SQL procedures that the menu items invoke.

These are the basic steps in designing a SQL*Menu application:

1. Analyze your business needs and plan the application.
2. Define the application objects:
   - application
   - menus
   - menu items
   - parameters, if any
   - PL/SQL procedures, if any
3. Save the application definition in the database.
4. Generate an application library file.
5. Run the application to test the menu items.

You should save your application every time you modify it. You should also print application documentation frequently as you build up your application, to keep track of the changes you make.

When you define the menu items, you specify the roles that can use each item. Your SQL*Menu DBA must create these roles.

In planning the application, you should consider the following issues:

- What will be the default menu display style?
- Will application operators use the letter selection method in pull-down or bar style menus? (If so, you need to plan the item names carefully.)
- What roles will have access to the application? Which menu items will they use? What options and help information do they need?
- What forms or other software products will the application invoke, or will a SQL*Forms application invoke the menu application?
- What background menu items would be most useful to the operators?
- What other applications will the operators be using?

Starting and Quitting SQL*Menu (Design)

Before you can use SQL*Menu (Design), you must be authorized to design menu applications. For information about authorization, see "Requirements for Using SQL*Menu" in Chapter 1 and "Security Operations" later in this chapter.

Logging In to SQL*Menu (Design)

The following steps describe how to start SQL*Menu (Design):

1. Enter `SQLMENU` at the operating system prompt.

   The SQL*Menu (Design) login screen appears. If you do not already have a list of the function key mappings for your operating system, make a note of the keyboard mapping for [Show Keys].

2. Type your username and press [Next Field].
3. Type your password and press [Next Field].

To bypass the login screen, you can enter `SQLMENU` *`username/Password`* at your operating system's prompt.

Other command options let you specify which terminal definition to use, specify input and output files, or display the command syntax. See "SQL*Menu System Components" in Chapter 12 for a full description of the command syntax.

Figure 5-1 shows the SQL*Menu (Design) login screen.

FIGURE 5-1
Login Screen for SQL*Menu (Design)

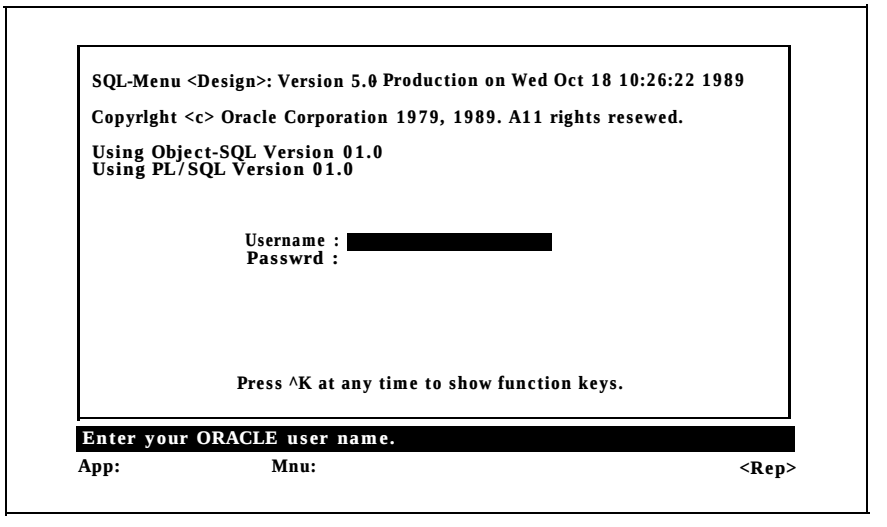
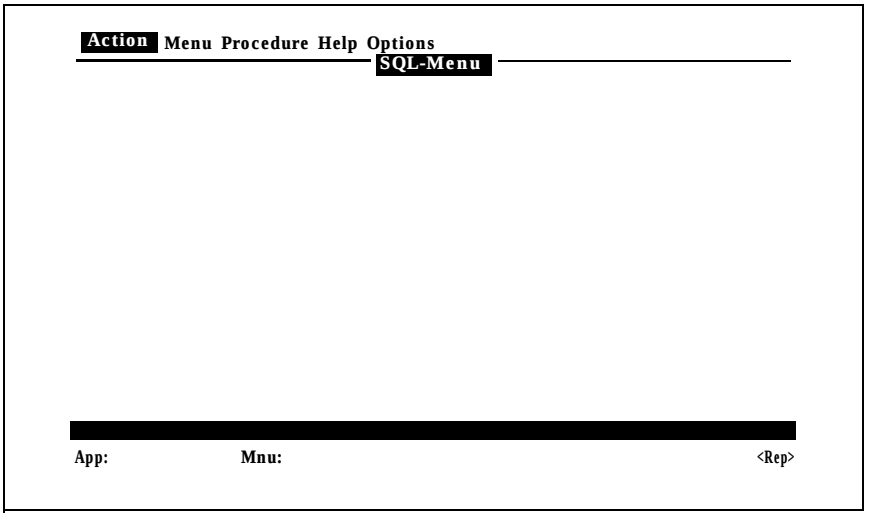


Figure 5-2 shows the SQL*Menu (Design) main menu that appears after you enter the login information.

FIGURE 5-2
SQL*Menu (Design) Main Menu

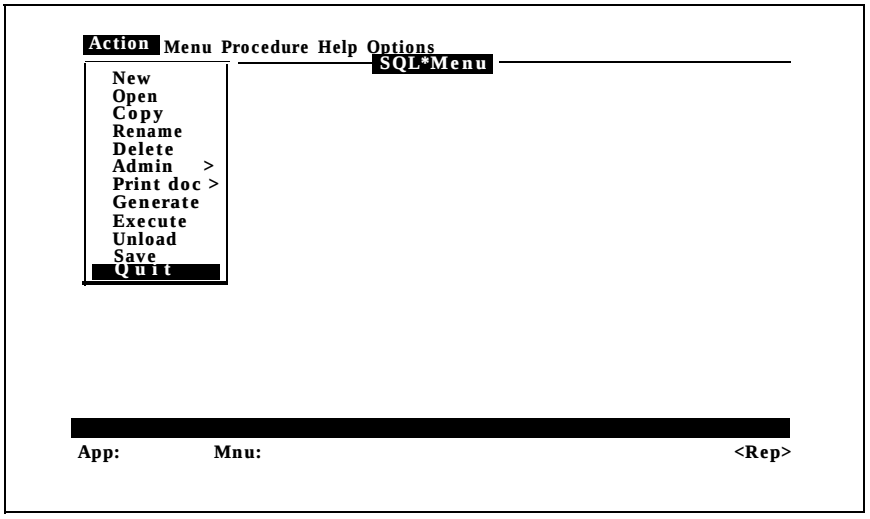


Quitting SQL*Menu (Design)

The following steps describe how to exit SQL*Menu (Design):

1. In the main menu of the design interface, select the Action item.
2. Select the Quit item from the Action submenu (Figure 5-3).

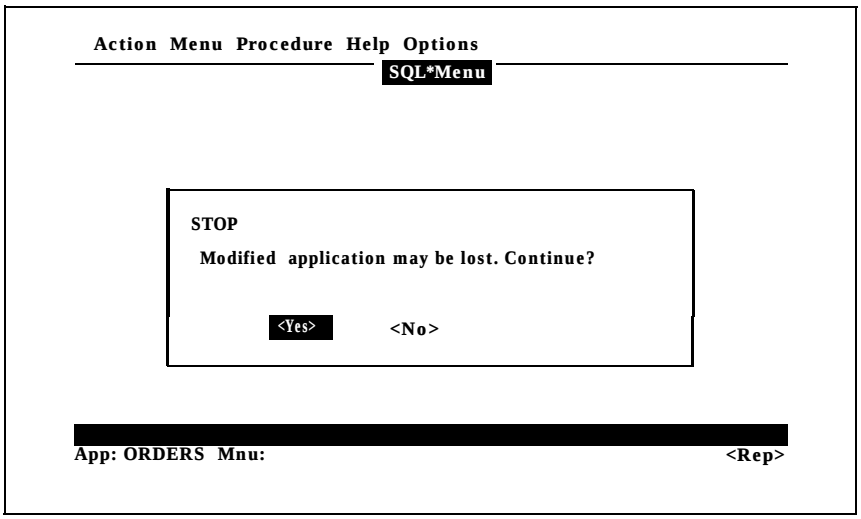
FIGURE 5-3
The Action Submenu in SQL*Menu (Design)



If you have modified the current application an alert appears when you try to quit (Figure 5-4), giving you a chance to stay in SQL*Menu (Design) and save your application before quitting.

3. To continue without saving your application, leave the cursor on the Yes button and press [Select].
4. To return to the main menu, move the cursor to the NO button and press [Select].

FIGURE 5-4
Alert for the Quit Item



Basic Design Operations

This section explains how to perform the basic operations needed for designing a menu application.

- find help information
- create a new application
- open an application
- modify application information
- define a menu
- define a menu item
- define a substitution parameter
- create a PL/SQL procedure
- copy or reference an object
- save an application
- generate an application
- execute an application
- print application documentation

Getting Help Information

Several kinds of online help are available to designers of SQL*Menu applications:

- hints and error messages in the message line
- function key help
- context-sensitive access to the online help system
- general access to the online help system

The online help system for SQL*Menu (Design) gives detailed help and general information about SQL*Menu and the design interface. You can obtain context-sensitive help information by pressing [Help], or you can select the Help System item from the Help submenu to display the introduction, “About the Online Help System.”

Once you have entered the online help system by either of these methods, you can access all other parts of it, as described in ‘The Online Help System” in Chapter 6.

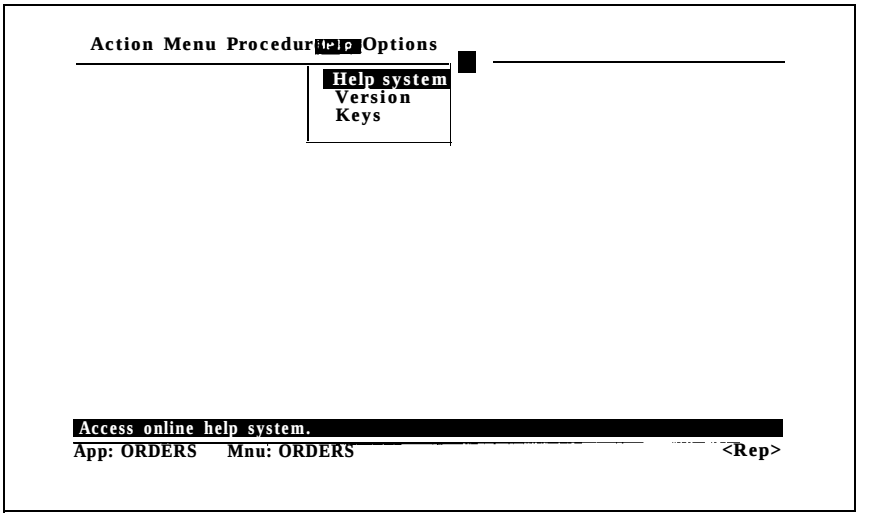
The Help submenu contains a Version item that displays information about the version of SQL*Menu that you are using, and a Keys item that displays the function key mappings. Selecting the Keys item in the Help submenu has the same effect as pressing [Show Keys].

To enter the online help system through the Help submenu, follow these steps:

1. In the main menu of the design interface, select the Help item.

   The Help submenu appears (Figure 5-5).

FIGURE 5-5
The Help Submenu in SQL*Menu (Design)



2. Select the Help System item from the Help submenu.

   The first screen of introductory text in the online help system appears.

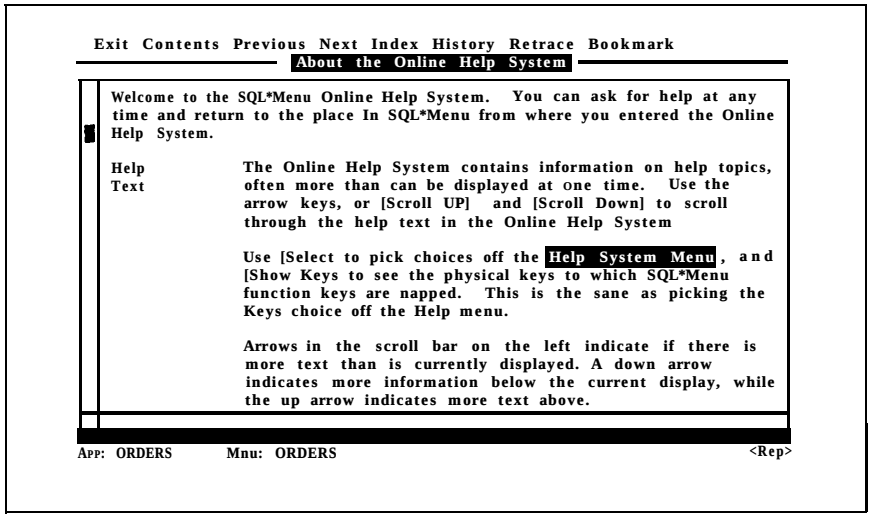
3. Read the introductory text to learn how to find information and navigate through the online help system.

4. To exit the online help system, press [Menu] and select the Exit item from the help system's main menu.

   SQL*Menu returns to the main menu of the design interface.

Figure 5-6 shows the first screen of the online help system.

FIGURE 5-6
The Online Help System



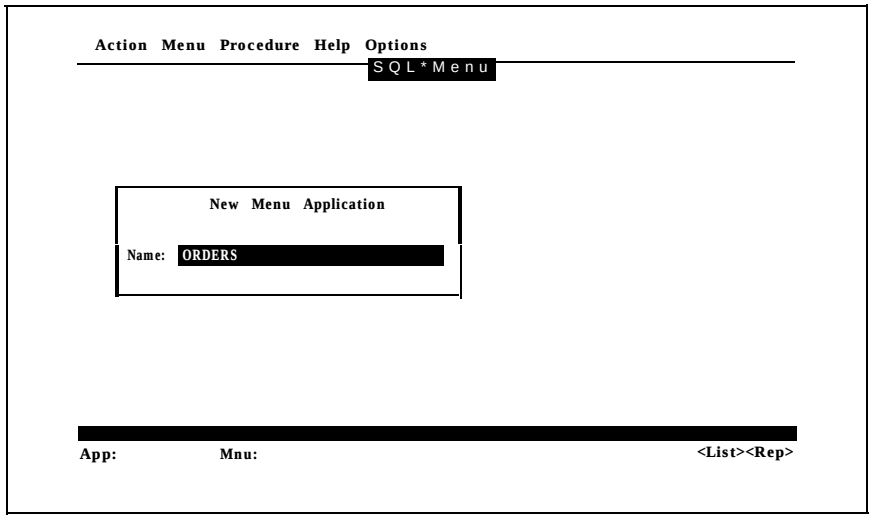
Creating a New Application

The following steps describe how to create a new menu application:

1. In the main menu of the design interface, select the Action item.
2. Select the New item from the Action submenu (Figure 5-3).

   The New Menu Application dialog box appears (Figure 5-7).

FIGURE 5-7
New Menu Application Dialog Box



3. In the Name field, enter a name for the application you want to create.

   Note: The name of your new menu application cannot be the name of an application that already exists in the database. If you enter an existing name, the message line displays an error message.

   To display a list of all existing applications in the database, press [List] while the cursor is in Name field.

4. Press [Accept] to create the new application.

   SQL*Menu initializes the application and returns to the main menu.

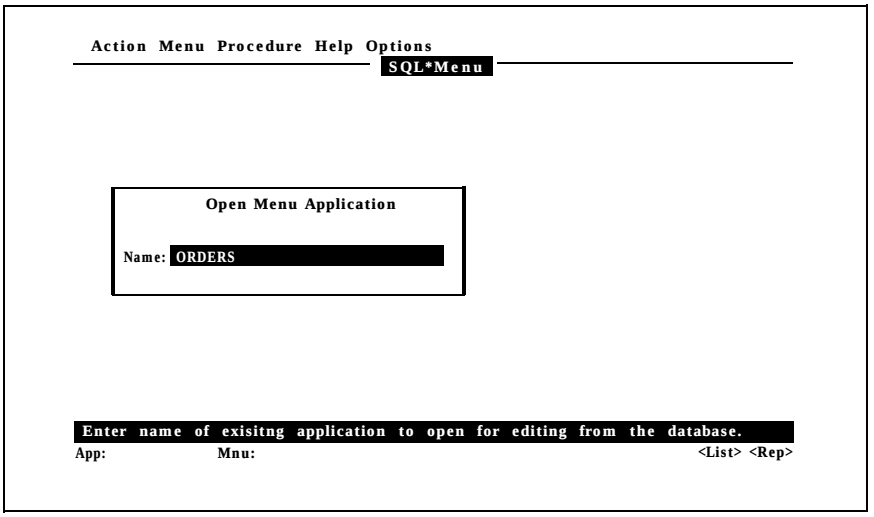
Opening a Menu Application

The following steps describe how to open an existing menu application:

1. In the main menu of the design interface, select the Action item.
2. Select the Open item from the Action submenu.

   The Open Menu Application dialog box appears (Figure 5-8).

FIGURE 5-8
Open Menu Application Dialog Box



3. Enter the application name in the Name field of the Open Menu Application dialog box (Figure 5-9).

   To select an application name from a list of values, press [List]. (See “List of Values” in Chapter 6 for more information.)

4. Press [Accept].

   SQL*Menu retrieves the application information from the database and ret-urns to the main menu.

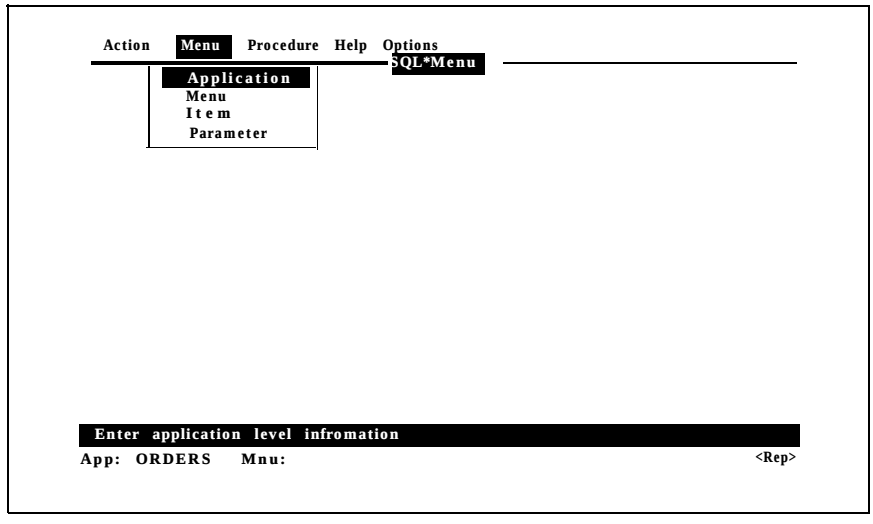
Modifying an Application Definition

The following steps describe how to view and modify the application definition for the current application:

1. In the main menu of the design interface, select the Menu item. (Before you can select the Menu item in the main menu, you must open an application.)

   The Menu submenu appears (Figure 5-9).

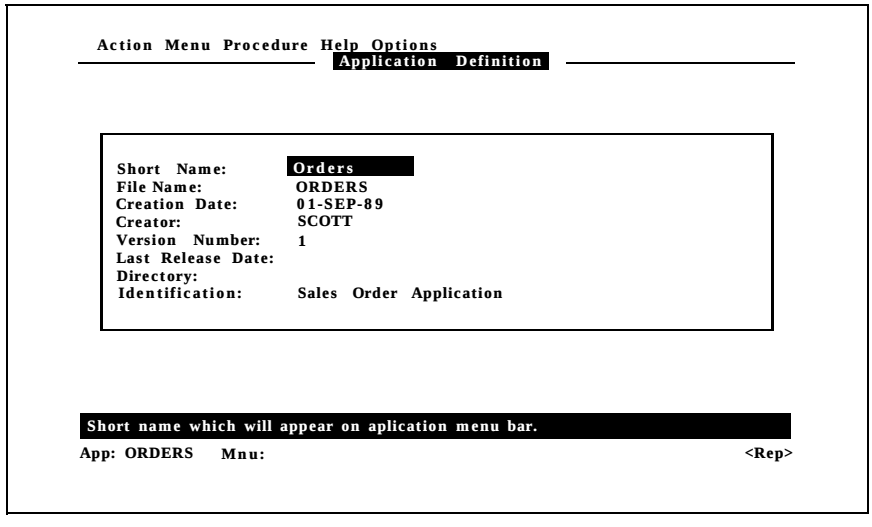
FIGURE 5-9
The Menu Submenu in SQL*Menu (Design)



2. Select the Application item from the Menu submenu.

   The Application Definition form appears (Figure 5-10).

FIGURE 5-10
Application Definition Form



The Application Definition form defines general information about the application, such as its creation date and file name. Table 5-1 describes how to fill in the elements of the Application Definition form.

3. To change the application information, type new values in the fields of the Application Definition form and press [Accept].

   SQL*Menu closes the Application Definition form and returns to the main menu.

TABLE 5-1
Application Definition Form

Interface Element	*Instructions*
Short Name (field)	Enter a short name for use in bar and pull-down menu display styles. The default value is the current menu application name, truncated to 15 characters if necessary.
File Name (field)	Enter a name for the file containing the application. Do not enter the file extension; SQL*Menu adds the extension when it creates *filename.* DMM. The default value is the current application name.
Creation Date (field)	Enter the date that the application was created, in the form DD-MON-YY. The default value is the current date.
Creator (field)	Enter the username of the creator or modifier. The default value is the current ORACLE user.
Version Number (field)	Enter the version and release number.
Last Release Date (field)	Enter the date of the application's last release, in the form DD-MON-YY. This field is optional.
Directory (field)	Enter the directory path for storing the library file. The default value is the current user's default directory.
Identification (field)	Describe the application. This 40-character identification appears in the application menu, either in the message line (pull-down and bar display styles) or next to the application number (full-screen display style). The default value is the application name.

For more information about defining a menu application, see "Applications" in Chapter 8.

Defining a Menu

The following steps describe how to define a menu for the current application:

1. In the main menu of the design interface, select the Menu item. (Before you can select the Menu item in the main menu, you must open an application.)

   The Menu submenu appears (Figure 5-9).

2. Select the Menu item from the Menu submenu.

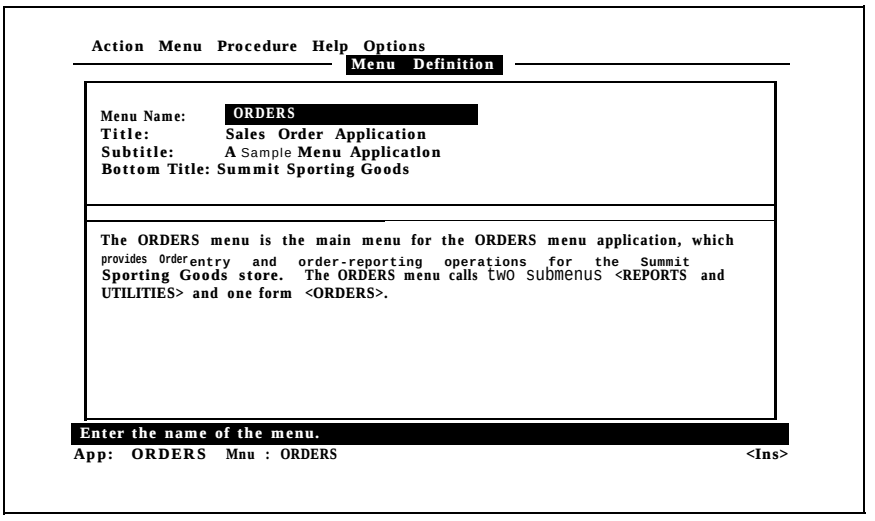
   The Menu Definition form or spread table appears.

3. If the spread table appears, press [Change Display Type] to display the form.

   The Menu Definition form displays information for one menu at a time. When you are familiar with the design interface, you might prefer using the Menu Definition spread table, which displays information for different menus in different rows, or records, of the spread table.

   Figure 5-11 shows the Menu Definition form for the main menu of the Orders Application. For an illustration of the Menu Definition spread table, see "Menus" in Chapter 8.

FIGURE 5-11
Menu Definition Form



4. If any menus already exist for the current application, press [Insert Record] to create a blank record for a new menu.

   SQL*Menu inserts a blank record and moves the cursor to it.

5. Fill in the menu definition information in the Menu Definition form as described in Table 5-2.

TABLE 5-2
Menu Definition Form or Spread Table

Interface Element	*Instructions*
Menu Name (field)	Enter a name of up to 30 characters for the menu. The menu name appears in the status line and identifies the menu for navigation. This field is required.
Title (field)	Enter a title of up to 40 characters. The title appears in full-screen displays in the title bar at the top of the menu. This field is required.
Subtitle (field)	Enter a subtitle of up to 40 characters. This field is required.
Bottom Title (field)	Optionally enter a bottom title of up to 72 characters. This text appears above the message line and status line.
Purpose (scroll region)	Describe the purpose of the menu and add design comments.

Note: The main menu, also known as the root menu, must have the same name as the application.

6. To create another menu, press [Insert Record] and define the menu in the blank record.

7. To delete a menu, move the cursor to the record for that menu and press [Delete Record].

8. Press [Accept] when you finish defining menus for the current application.

   The menu defined in the current record becomes the current menu (`Mnu:` in the status line) and SQL*Menu returns to the main menu of the design interface.

For more information about defining menus, see “Menus” in Chapter 8.

Defining a Menu Item

Before you can define a menu item, the menu that contains the item must become the current menu. The following steps describe how to define items for the current menu and to change the current menu:

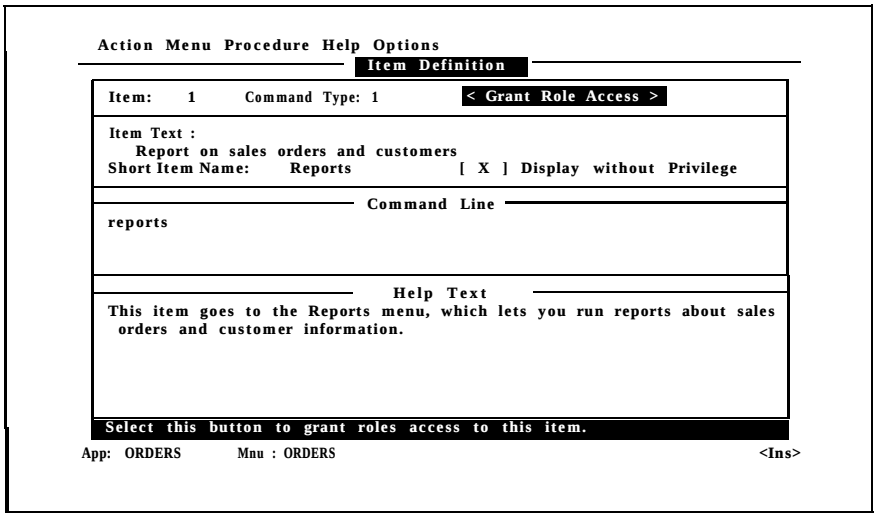
1. In the main menu of the design interface, select the Menu item. (&fore you can select the Menu item in the main menu, you must open an application.)
2. Select the Menu item from the Menu submenu.

   The Menu Definition form or spread table appears.
3. If the form appears, press [Change Display Type] to display the spread table.

   The Menu Definition spread table displays information about different menus on different rows. The spread table is most useful when you want to see all the menus in an application at once.
4. In the Menu Definition spread table, press [Down] to move to the menu you want to make current.
5. Press [Zoom In] to make the menu current and move to the Item Definition form or spread table.

   Figure 5-12 shows the Item Definition form for the first item in the main menu of the Orders Application. For an illustration of the Item Definition spread table, see "Menu Items" in Chapter 8.

FIGURE 5-12
Item Definition Form



Note: The status line always shows the current menu (for example, `Mnu: ORDERS` in Figure 5-12).

6. If any items already exist for the current menu, press [Insert Record] to create a blank record for a new menu item.

   SQL*Menu inserts a blank record and moves the cursor to it.

7. Create a menu item for the current menu by filling in the information in the Item Definition form, as described in Table 5-3.

TABLE 5-3
Item Definition Form or Spread Table

Interface Element	*Instructions*
Item (field)	Enter an item number to specify the display order within the menu. The default value is one greater than the previous item number.
Command Type (field)	Enter a number for the type of command statement in the Command Line: 1 Invoke a submenu 2 Execute an operating system command 3 Execute an operating system command, then pause for an operator response 4 Invoke SQL*Forms (Run Form) 5 Invoke SQL*Plus 6 Execute a SQL*Menu macro command 7 Execute a PL/SQL command
Grant Role Access (button)	Select this button to display the Item Role window, then enter the names of existing roles that need access to this menu item.
Item Text (field)	Enter a description of up to 70 characters. This text appears in the message line for pull-down and bar style displays, or next to the item number in full-screen displays.
Short Item Name (field)	Enter a descriptive name of up to 15 characters. This name appears as the menu item in pull-down and bar style displays.
Display without Privilege (check box)	Select this check box to display the item to users who do not have access to use it. (Inaccessible items appear without highlighting.) The default value is off (not displaying items to unauthorized users).
Command Line (scroll region)	Enter a menu name, macro, or command statement that matches the specified Command Type.
Help Text (scroll region)	Enter the help text for this menu item. This help text appears if the operator presses [Help] when the cursor is on this item.

"Menu Items" in Chapter 8 describes the different types of commands you can use in the Command Line. For example, to have an item display a submenu, you enter the menu name in the Command Line and type 1 in the Command Type field.

8. To make another menu current, press [Zoom Out] and move to the menu you want, then press [Zoom In].
9. Create items for the current menu by repeating steps 6 and 7.
10. To delete a menu item, move to the record for that item and press [Delete Record].
11. Press [Accept] when you finish defining menu items.

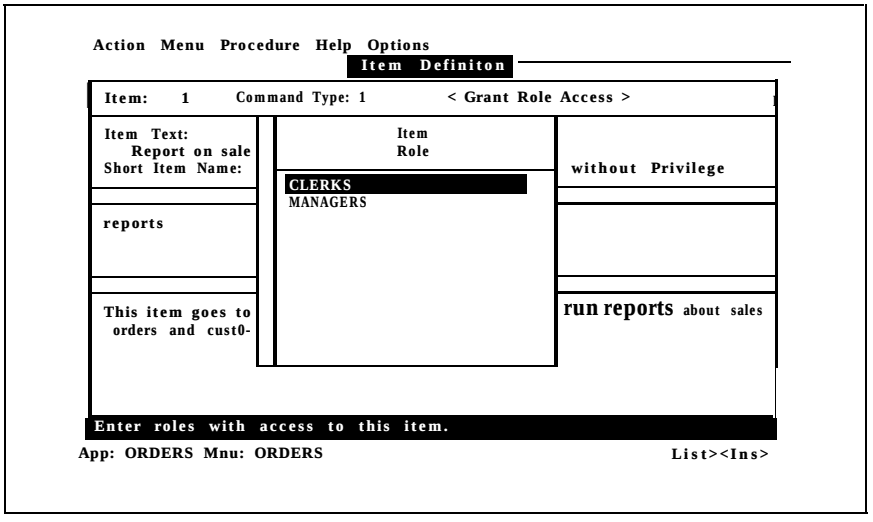
    SQL*Menu returns to the main menu of the design interface.

Granting Role Access

Figure 5-13 shows the Item Role window that appears when you select the Grant Role Access button. To give a role access to the current menu item, enter the role name in this window. Press [Insert Record] to add a blank record for a new role name, or press [Delete Record] to revoke the current role's access to the item.

Note: If you do not explicitly assign any roles to a menu item, all roles that have access to the application have access to the item.

FIGURE S-13
Item Role Window



Adding Help Text

The Help Text for an item appears when you run the application and press [Help] with the cursor on that item. You can add the help text when you define the menu item, or any time after defining the item.

Defining a Substitution Parameter

A substitution parameter is a variable that can be used to stand for a value in the command line of a menu item or in a PL/SQL procedure. SQL*Menu substitution parameters have two-character names preceded by an ampersand (&) or, in PL/SQL code, a colon (:).

For example, the five predefined substitution parameters are called &AD, &PW, &SO, &TT, and &UN in the command line for an item; but in a PL/SQL procedure they are called :AD, :PW, :SO, :TT, and :UN. For descriptions of these predefined parameters, see "Substitution Parameters" in Chapter 8.

The following steps describe how to create a substitution parameter for the current application:

1. Select the Menu item from the main menu of the design interface.

   The Menu submenu appears (Figure 5-14).

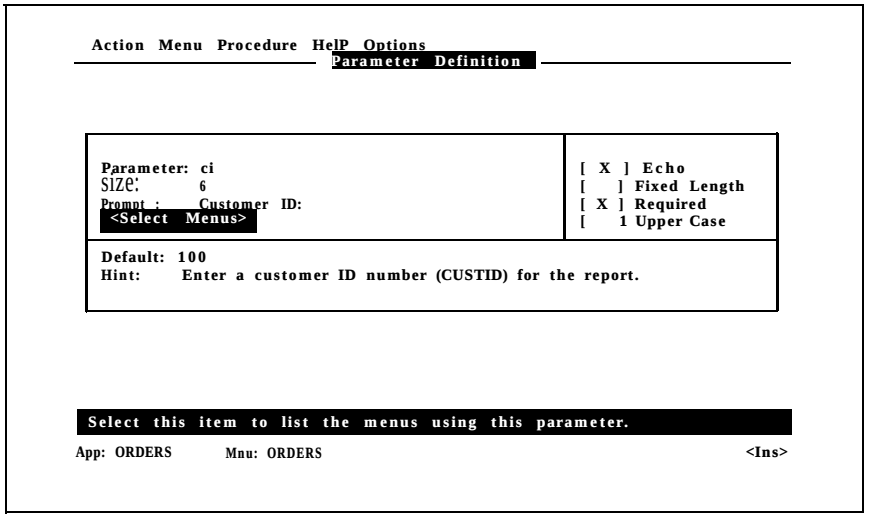
2. Select the Parameter item from the Menu submenu.

   The Parameter Definition form or spread table appears.

3. If the spread table appears, press [Change Display Type] to display the form.

   Figure 5-14 shows the Parameter Definition form displaying the &CI parameter (customer identification) in the Orders Application. For an illustration of the Parameter Definition spread table, see "Substitution Parameters" in Chapter 8.

FIGURE 5-14
Parameter Definition Form



4. If any parameters already exist for the current application, press [Insert Record] to create a blank record for a new parameter.
5. Fill in the two-character parameter code and the rest of the information that defines-a substitution parameter in the Parameter Definition form, as described in Table 5-4.

   To toggle the settings of the check boxes, press [Select].

TABLE 5-4
Parameter Definition Form and Spread Table

Interface Element	*Instructions*
Parameter (field)	Enter a two-character abbreviation for the substitution parameter.
Size (field)	Specify the maximum number of characters the user can enter for the value of the parameter.
Prompt (field)	Enter the text that will prompt a user to supply a value for the substitution parameter.
Select Menus (button)	Press [Select] to display the Menu Name window, then enter the names of menus that you want to associate with the parameter in the full-screen display style.
Echo (check box)	Type X to display the substitution parameter on your terminal screen as you type it. Type a blank to suppress the display.
Fixed Length (check box)	Type X or a blank to specify whether the user must always fill the parameter size by entering the maximum number of characters allowed.
Required (check box)	Type X or a blank to specify whether you must always supply a value for the parameter.
Upper Case (check box)	Type X or a blank to specify whether or not the value entered for the substitution parameter should be converted to upper case.
Default (field)	Specify the parameter's default value.
Hint (field)	Describe the type of parameter value an operator should enter and list some possible values.

6. To create another parameter, press [Insert Record] and define the parameter in the blank record.
7. To delete a parameter, move the cursor to the record for that parameter and press [Delete Record].
8. Press [Accept] when you finish defining parameters for the current application.

   SQL*Menu returns to the main menu of the design interface.

For more information about defining and using parameters, see "Substitution Parameters" in Chapter 8 and "Referencing Objects in PL/SQL" in Chapter 9.

Associating a Substitution Parameter with a Menu Item

To associate a substitution parameter with a menu item, use the parameter in that item's command line when you define the item in the Item Definition form or spread table.

When an operator selects a menu item associated with a substitution parameter, the Enter Parameter Values dialog box appears and prompts the operator for the parameter value.

An ampersand (&) must precede the two-character parameter name in the command sting. In PL/SQL blocks, a colon (:) precedes the parameter name instead of an ampersand.

Associating a Substitution Parameter with a Menu

In the full-screen menu display style, a substitution parameter can be associated with an entire menu. When an operator navigates to that menu, the Enter Parameter Values dialog box appears, prompting for the parameter value. The value entered is then available to all items in that menu.

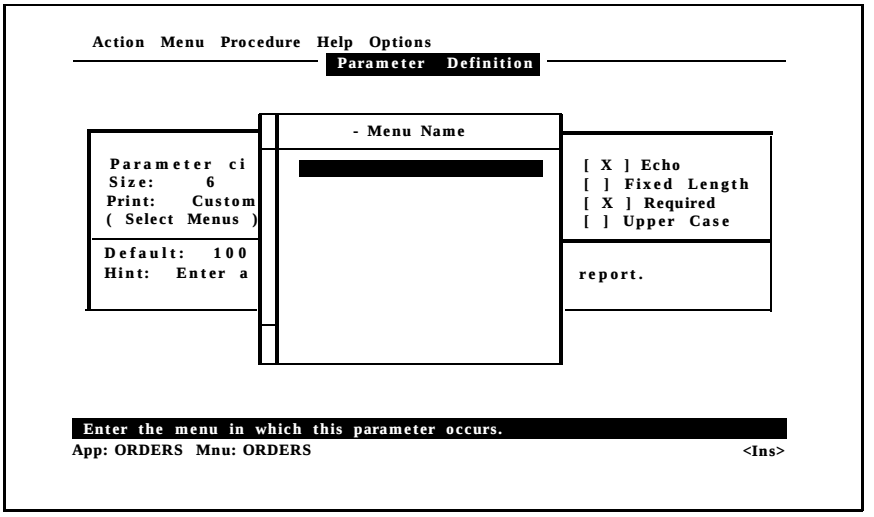
The following steps describe how to associate a substitution parameter with a full-screen menu in the current application:

1. Display the Parameter Definition form or spread table as described earlier in this section.
2. Press [Down] as many times as needed to move to the record for the parameter that you want to turn into a menu parameter.
3. Press [Next Field] three times to move the cursor to the Select Menus button.
4. Press [Select].

   The Menu Name window appears.

Figure 5-15 shows the Menu Name window that appears when you select the Select Menus button in the Parameter Definition form.

FIGURE 5-15
The Menu Name Window



5. Enter the name of the menu you want to associate with the parameter. To use the parameter with more than one menu, you must list the name of each menu to which it applies.
6. To associate another menu with the current parameter, press [Insert Record] and enter the menu name in the blank record.
7. To delete a menu from the Menu Name window, move the cursor to that menu name and press [Delete Record].
8. Press [Accept] to close the Menu Name window.

   SQL*Menu returns to the Parameter Definition form or spread table.
9. To associate menus with another parameter, press [Up] or [Down] to move to the record for that parameter, then repeat steps 2 to 7.
10. Press [Accept] in the Parameter Definition form or spread table when you finish associating menus with substitution parameters for the current application.

    SQL*Menu returns to the main menu of the design interface.

Note: In pull-down and bar style menus, a menu parameter is treated as an item parameter: the Enter Parameter Values dialog box appears when an operator selects an item associated with that parameter.

Defining a PL/SQL Procedure

The following steps describe how to define a named procedure for the current application.

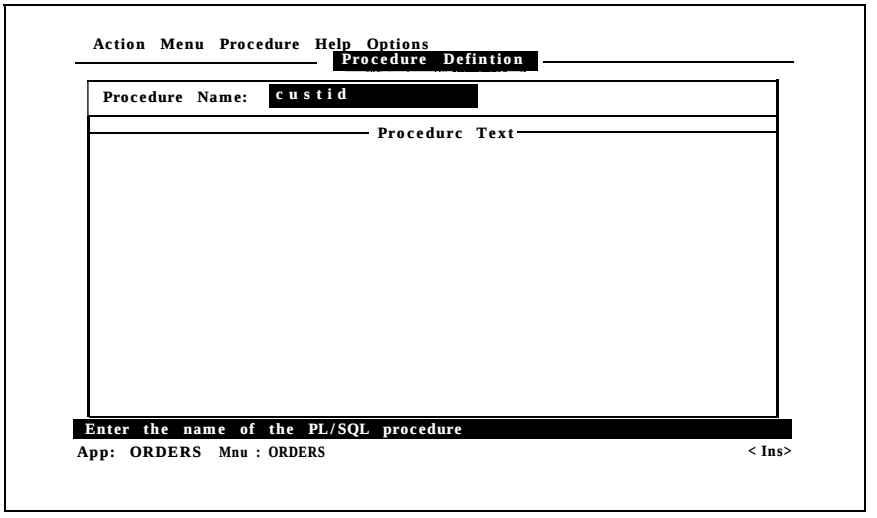
1. Select the Procedure item from the main menu of the design interface.

   The Procedure Definition form or spread table appears.

2. If the spread table appears, press [Change Display Type] to display the form.

   Figure 5-16 shows the Procedure Definition form. For an illustration of the Procedure Definition spread table, see 'Procedures" in Chapter 8.

FIGURE 5-16
Procedure Definition Form



3. If any procedures already exist for the current application, press [Insert Record] to create a blank record for a new procedure.
4. Fill in the procedure name and text, as described in Table 5-5.

TABLE 5-5
Procedure Definition Form and Spread Table

Interface Element	*Instructions*
Procedure Name (field)	Enter a name to identify the procedure, following ORACLE naming conventions.
Procedure Text (scroll region)	Enter the PL/SQL block for the procedure in this scroll region. (For information about writing PL/SQL blocks, see Chapter 9, "PL/SQL.")

5. To create another procedure, press [Insert Record] and define the procedure in the blank record.
6. To delete a procedure, move the cursor to the record for that procedure and press [Delete Record].
7. Press [Accept] when you finish defining procedures for the current application.

   SQL*Menu returns to the main menu of the design interface.

Using a PL/SQL Procedure

To make a menu item call a PL/SQL procedure, enter 7 in the . Command Type field of the Item Definition form or spread table, and enter the procedure name in the Command Line scroll region.

Note: Command type 7 also lets you enter text for an unnamed procedure in the Command Line scroll region of the Item Definition form or spread table, as described below.

Defining an Unnamed Procedure

An *unnamed procedure* is a block of PL/SQL code ("anonymous block") that can only be invoked by selecting the associated menu item. An unnamed procedure can call PL/SQL packaged procedures and other named procedures that were created by application designers, but it cannot be called by other procedures.

To make a menu item execute an unnamed procedure, enter 7 in the Command Type field of the Item Definition form or spread table, and enter the procedure text in the Command Line scroll region.

If you want to use an unnamed procedure in more than one menu item, you can make a copy of it by copying the entire item to a new context. To copy an item, press [Copy Object] and use the Copy/Reference Object dialog box as described in the following section.

After you copy the item, you might have to modify some of its characteristics. For example, you can change the order of items within a menu by altering the numbers in the Item field of the Item Definition form or spread table.

For more information about defining and using PL/SQL procedures, see "Procedures" in Chapter 8 and "PL/SQL Blocks in SQL*Menu" in Chapter 9.

Copying or Referencing an Object

The Copy/Reference Object dialog box allows you to copy or reference an object from any SQL*Menu application to the current application. The object can be a menu, item, parameter, or procedure.

Copying an object creates an independent new object that you can modify in its new location. Referencing an object creates a duplicate object that you cannot modify except by changing the original object.

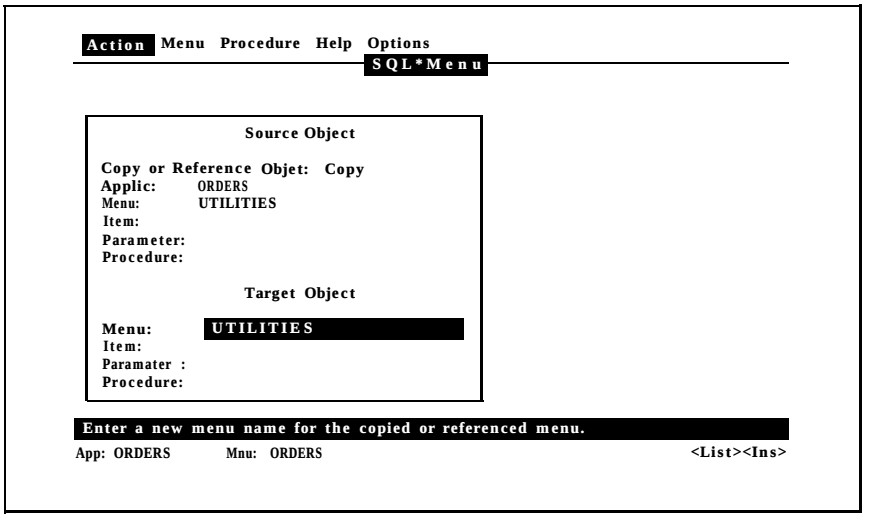
Note: You cannot reference an object within the application in which it is defined, but you can copy objects within an application.

The following steps describe how to copy or reference an object:

1. In the design interface, press [Copy Object].

   The Copy/Reference Object dialog box appears (Figure 5-21).

FIGURE 5-17
Copy/Reference Object Dialog Box



2. In the Copy or Reference Object field, either enter Copy to make an independent copy of the objector enter Reference to make a copy that refers to the original object.
3. Enter the name and location of the object you want to copy or refer to in the Source Object fields.
4. Enter the new name and location in the Target Object fields.
5. Press [Accept] or [Next Field] to make the copy or reference.

   SQL*Menu creates the copy or reference and returns to the main menu.

Saving an Application

The object definitions that you create or modify in the design interface must be saved in the database before you exit SQL*Menu. If you do not save the application in the database, you will lose the object definitions that you created or modified during the current session in the design interface.

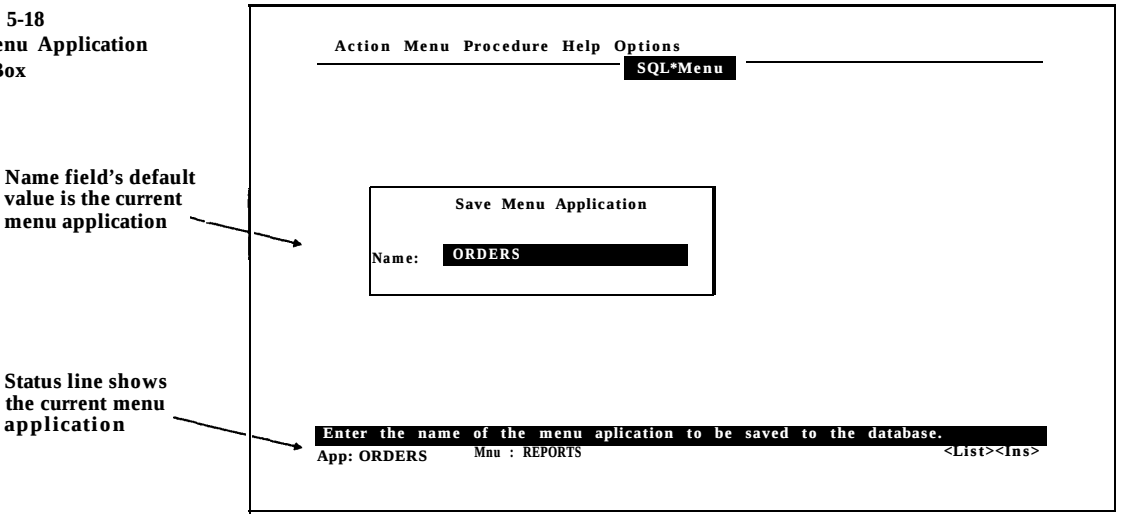
When you first create an application, and whenever you modify the directory specified in the Application Definition form, you must save the application before you generate the library file.

The following steps describe how to save the object definitions for an application to the database:

1. Select the Action item from the main menu of the design interface.
2. Select the Save item from the Action submenu.

   The Save Menu Application dialog box appears (Figure 5-18) with the name of the current application in the Name field.

FIGURE 5-18
Save Menu Application Dialog Box



3. Press [Accept] to save the current application, or enter another application name and press [Accept]. (Press [List] to display a list of values for the application name.)

   SQL*Menu displays the `Working....` message, then returns to the main menu of the design interface.

Generating an Application

Before you can run a menu application, you must generate it. When you generate a menu application, SQL*Menu creates a library file, *filename.* DMM, using the file name and the directory path specified in the Application Definition form.

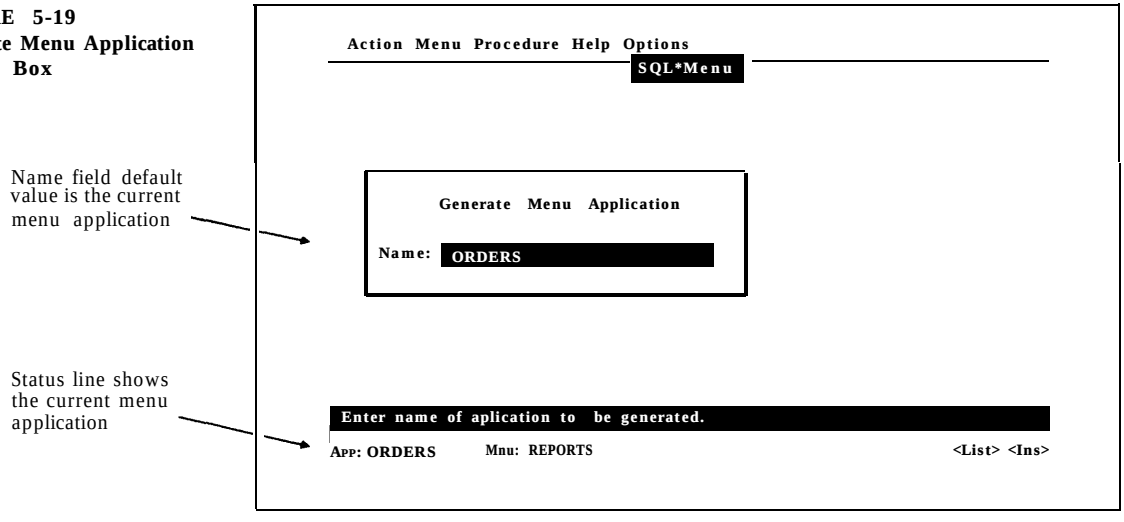
Note: Generating an application does not save it in the database. Remember to save the application before you exit SQL*Menu.

The following steps describe how to generate a menu application:

1. Select the Action item from the main menu of the design interface.
2. Select the Generate item from the Action submenu.

   The Generate Menu Application dialog box appears (Figure 5-19) with the name of the current menu application in the Name field. To display a list of values for the application name, press [List].

FIGURE 5-19
Generate Menu Application Dialog Box



3. Press [Accept] to generate the current application, or enter another application name and press [Accept].

   SQL*Menu displays the `Working... .` message, then `Press any key to continue...`, and returns to the main menu when you press a key.

The Generate before Executing Menu Option

If the Generate before Executing Menu option is turned on in the . Option Selection form, you can execute an altered application without first selecting the Generate item in the Action menu.

For more information about SQL*Menu options, see "Setting Options in SQL*Menu (Design)" later in this chapter.

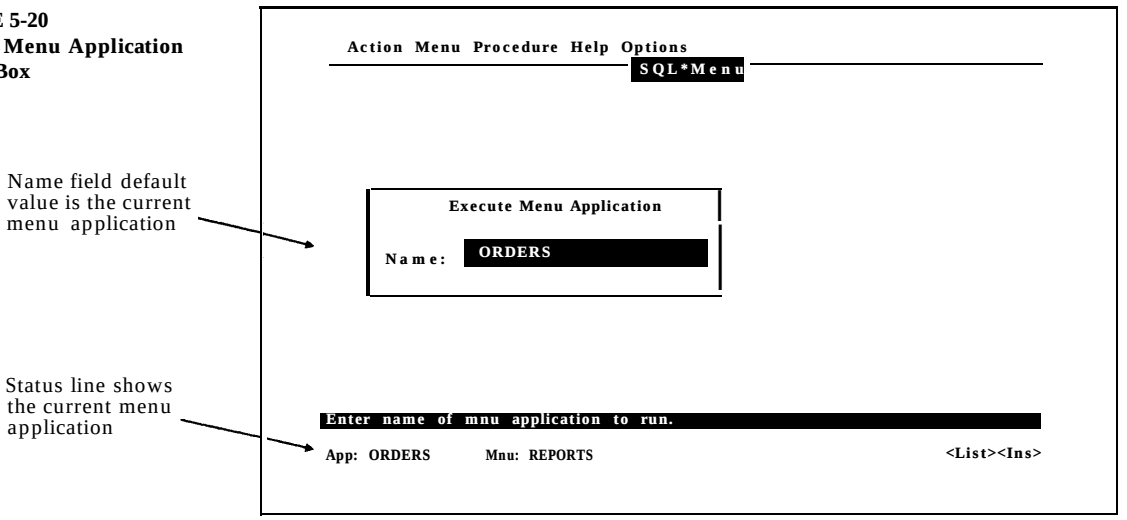
Executing a Menu Application

The following steps describe how to run a menu application from SQL*Menu (Design):

1. Select the Action item from the main menu of the design interface.
2. Select the Execute item from the Action submenu.

   The Execute Menu Application dialog box appears (Figure 5-20) with the name of the current menu application in the Name field.

FIGURE 5-20
Execute Menu Application Dialog Box



3. Press [Accept] to execute the current application, or enter another application name and press [Accept]. (Press [List] to display a list of values for the application name.)

   SQL*Menu runs the menu application, starting at the main menu. When you exit the application, you return to SQL*Menu (Design).

The Menu Display Style Option

Before you execute an application from the design interface, you can specify a menu display style by selecting one of the radio buttons in the Option Selection form. (For information about radio buttons in the design interface, see "Buttons and Radio Buttons" in Chapter 6.)

The Debug Mode Option

When you run an application to test your menu items, you can display the command line currently being executed by pressing [Debug Mode]. To turn off the debug mode option, press [Debug Mode] again.

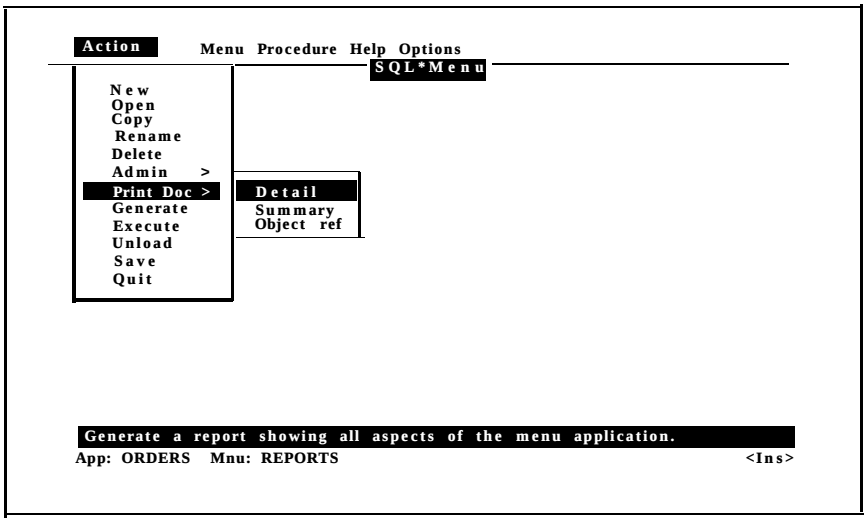
For more information about SQL*Menu options, see "Setting Options in SQL*Menu (Design)" later in this chapter.

Printing Application Information

The following steps describe how to list the object definitions for a menu application or referenced object, either in a file or on a printout:

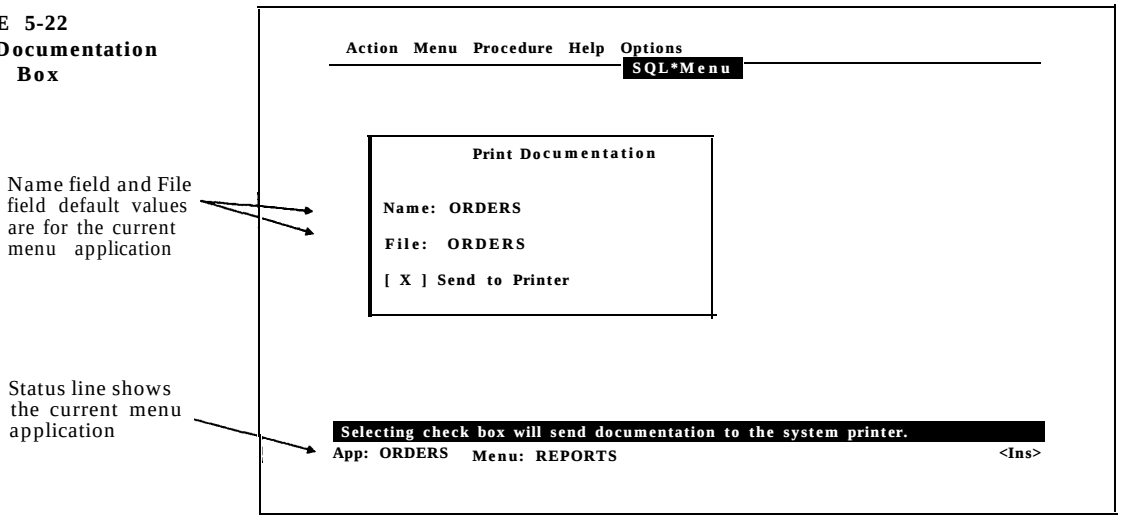
1. Select the Action item from the main menu of the design interface.
2. Select the Print Doc item from the Action submenu.
3. For complete application documentation, select the Detail item from the Print Documentation submenu (Figure 5-21).

FIGURE 5-21
The Print Documentation Submenu



The Print Documentation dialog box appears (Figure 5-22).

FIGURE 5-22
Print Documentation Dialog Box



The Print Documentation dialog box has as default values the name of the current menu application in the Name field and the filename for the current application in the File field.

4. To print the complete documentation on the default system printer, move the cursor to the Send to Printer check box and press [Select], or type any character except a blank.

5. Press [Accept] or [Next Field] to create a documentation file named *filename.* DOC in the default directory, where *filename* by default is the application filename. (To specify another filename, type the filename in the File field before you press [Accept].)

   SQL*Menu creates the documentation file, and prints a copy of it if you specified Send to Printer, then returns to the main menu of the design interface.

6. For a brief listing that omits the help text, perform steps 1 through 5 but instead of step 3, select the Summary item from the Print Documentation submenu (Figure 5-25).

7. For documentation about references to a specific object, perform steps 1 through 5 but instead of step 3, select the Object Ref item from the Print Documentation submenu. In the Print Documentation dialog box, enter the complete context for the object you want to document:

   - application name
   - menu name
   - item name
   - parameter name
   - procedure name

   For an explanation of object context, see “Object Hierarchy” in Chapter 8.

Setting Options in SQL*Menu (Design)

The SQL*Menu (Design) Option Selection form lets you turn the following SQL*Menu options on or off during a session in the design interface

Use Forms as Default This option selects forms, rather than spread tables, as the default display style in the design interface.

Show List of Values This option automatically displays a list of values whenever one is available in the current context of the design interface.

Generate before Executing Menu This option automatically generates a new library file for a menu application before executing the application.

Suppress Hints This option conceals the hint messages that normally appear in the message line. Error messages are not suppressed by this option.

Disable PL/SQL Compilation This option turns off the automatic compilation of PL/SQL procedures in the design interface.

Auto Restrict List of Values This option automatically restricts a list of values to values that match a search pattern based on the value in the corresponding field.

Show Detailed Working Messages This option displays detailed messages, instead of the `Working . . .` message, when SQL*Menu is processing a command in the design interface.

Menu Display Style This set of radio buttons selects the default menu display style to use when running applications from the design interface. (You can also specify a default menu display style with a command line switch when you use the RUNMENU command to run an application, as described in "SQL*Menu System Components" in Chapter 12.)

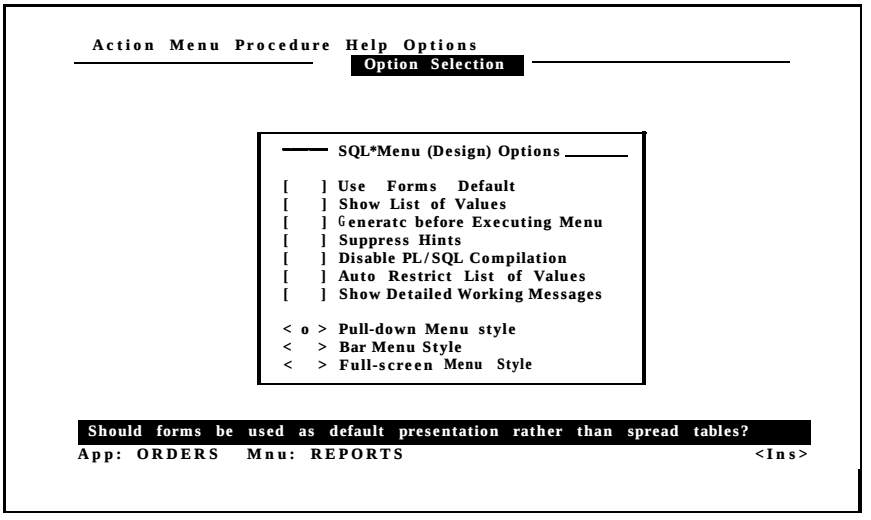
The SQL*Menu (Design) Option Selection form contains check boxes and radio buttons for changing options. For complete descriptions of these interface elements, see Chapter 6, "Elements in the Design Interface."

The following steps describe how to change the options settings in the SQL*Menu (Design) Option Selection form

1. Select the Options item from the main menu of the design interface.

   The SQL*Menu (Design) Option Selection form appears, as shown in Figure 5-23.

FIGURE 5-23
SQL*Menu (Design) Option Selection Form



2. For each check box or button, press [Select] to change the setting as needed.
3. Press [Next Field] or [Previous Field] to move the cursor to another check box or to the set of radio buttons.
4. When you have finished changing the option settings, press [Accept].

   SQL*Menu makes the changes and returns to the main menu of the design interface.

User Preference File

You can create a user preference file that automatically enforces SQL*Menu options every time that you log in to SQL*Menu (Design). This file can contain keywords and settings that allow you to preset the following options:

- all of the SQL*Menu (Design) options
- a SQL*Menu (Design) mapping
- the SQL*Menu (Run Menu) menu style

For example, to ensure that SQL*Menu (Design) always runs without displaying hints, you would include the following line in the user preference file:

```
SUPPRESS_HINTS = ON
```

You can specify as many options as you want in the file, in any order.

To be in effect during a SQL*Menu (Design) session, the user preference file must be named SQLMENU.CFG and it must reside in the current directory.

If you log in to SQL*Menu (Design) with a command line switch that corresponds to a preset design option or mapping, that command line switch overrides the setting in the user preference file. Also, if a line in the user preference file contains an error, SQL*Menu ignores that line when it reads the file.

Presetting Design Options

To preset a SQL*Menu (Design) option, include the appropriate keyword and setting in the user preference file, using the following syntax:

```
keyword = [ON | OFF]
```

The following list presents the keywords for each of the available design options. These keywords correspond directly with the names of the SQL*Menu (Design) options.

- AUTO_RESTRICT_LIST_OF_VALUES
- DISABLE_PL/SQL_COMPILATION
- GENERATE_BEFORE_EXECUTING_MENU
- SHOW_DETAILED_WORKING_MESSAGES
- SHOW_LIST_OF_VALUES
- SUPPRESS_HINTS
- USE_FORMS_AS_DEFAULT

Note that although all options are turned off by default, you can explicitly turn an option off in the file.

Presetting a Design Mapping

To preset the SQL*Menu (Design) mapping, include the MAPPING keyword and mapping name in the user preference file, using the following syntax:

```
MAPPING = [filename:]mapping
```

Presetting a Menu Style

To preset the SQL*Menu (Run Menu) menu style, include the MENU_STYLE keyword and setting in the user preference file, using the following syntax:

```
MENU_STYLE = [BAR |FULL_SCREEN |PULL_DOWN]
```

Application Maintenance Operations

This section describes how to perform general maintenance operations on menu applications:

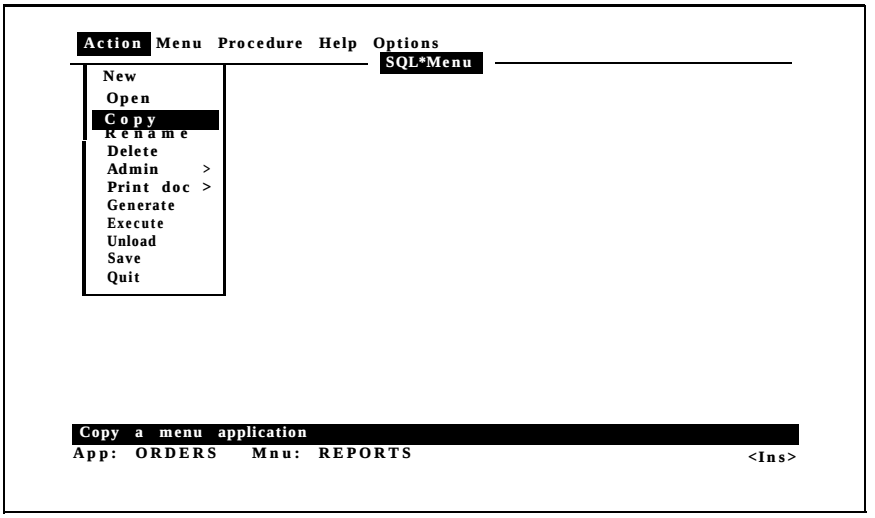
- copy an application
- rename an application
- delete an application
- export (unload) an application

Copying an Application

The following steps describe how to copy a menu application in the database:

1. Select the Action item from the main menu of the design interface.
2. Select the Copy item from the Action submenu (Figure 5-24).

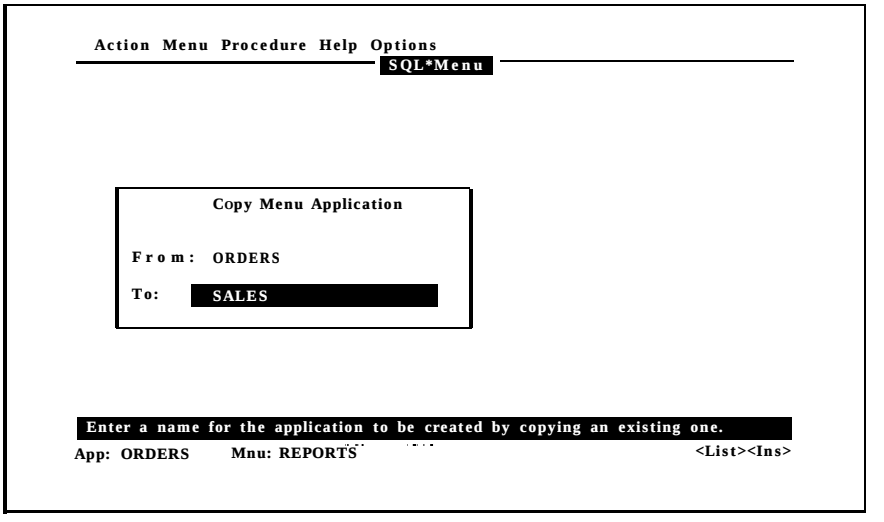
FIGURE 5-24
The Copy Item in the Action Submenu



The Copy Menu Application dialog box appears (Figure 5-25).

3. Enter the name of the application you want to copy in the From field. To select an existing application name from a list of values, press [List] when the cursor in the From field.
4. Enter a new name in the To field.

FIGURE 5-25
Copy Menu Application Dialog Box



5. Press [Accept] or [Next Field].

   SQL*Menu displays the `Working . . .` message while copying the application, then returns to the main menu.

6. Open the new copy of the menu application by selecting the Open item from the Action submenu and entering the application name.

7. Enter the Menu Definition form or spread table and change the name of the main menu to match the application name.

8. Enter the Application Definition form and change the application filename and short name to unique names.

9. Save the new application by selecting the Save item from the "Action submenu and pressing Return to accept the current application name in the Save Menu Application dialog box.

Note: Before you can run the new application, you must generate a library file for it. If you are going to-modify the copied application, you do not need to generate the library file until you have finished your modifications.

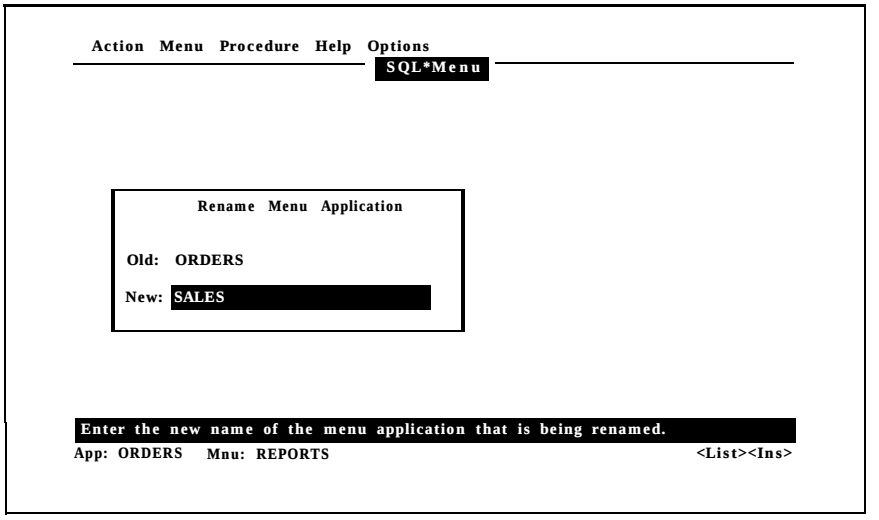
Renaming an Application

The following steps describe how to rename a menu application in the database:

1. Select the Action item from the main menu.
2. Select the Rename item from the Action submenu.

   The Rename Menu Application dialog box appears (Figure 5-26).

FIGURE 5-26
Rename Menu Application Dialog Box



3. Enter the name of the application you want to rename in the Old field. To select an existing application name from a list of values, press [List] when the cursor in the Old field.
4. Enter anew name in the New field.
5. Press [Accept] or [Next Field].

   SQL*Menu displays the `Working . . .` message while renaming the application, then returns to the main menu.
6. Open the renamed menu application by selecting the Open item from the Action submenu and entering the application name.
7. Enter the Menu Definition form or spread table and change the name of the main menu to match the new application name.
8. Optionally, enter the Application Definition form and change the application filename and short name.
9. Save the renamed application by selecting the Save item from the Action submenu and pressing Return to accept the current application name in the Save Menu Application dialog box.

Note: Before you can run the renamed application, you must generate a library file by selecting the Generate item from the Action submenu and entering the application name in the Generate Menu Application dialog box. If you want to modify the application, you do not need to generate the library file until you have finished your modifications.

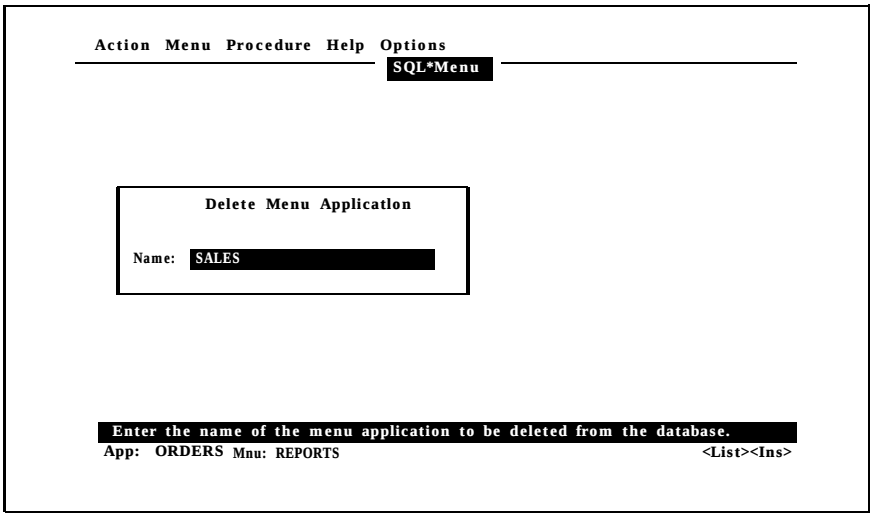
Deleting an Application

The following steps describe how to delete a menu application from the database:

1. Select the Action item from the main menu.
2. Select the Delete item from the Action submenu.

   The Delete Menu Application dialog box appears (Figure 5-27).

FIGURE 5-27
Delete Menu Application Dialog Box



3. Enter the name of the application you want to delete in the Name field. To select an existing application name from a list of values, press [List] when the cursor in the Name field.
4. Press [Accept] or [Next Field].

   SQL*Menu displays the `Working . . .` message while deleting the application, then returns to the main menu of the design interface.

Note: If a library file exists for the application you delete, you should also delete that library file. You can do this without logging out of SQL*Menu (Design) by using the Host item in the Admin submenu, which allows you to enter operating system commands. To return to SQL*Menu from the operating system, press Return on a blank line.

Unloading (Exporting) an Application

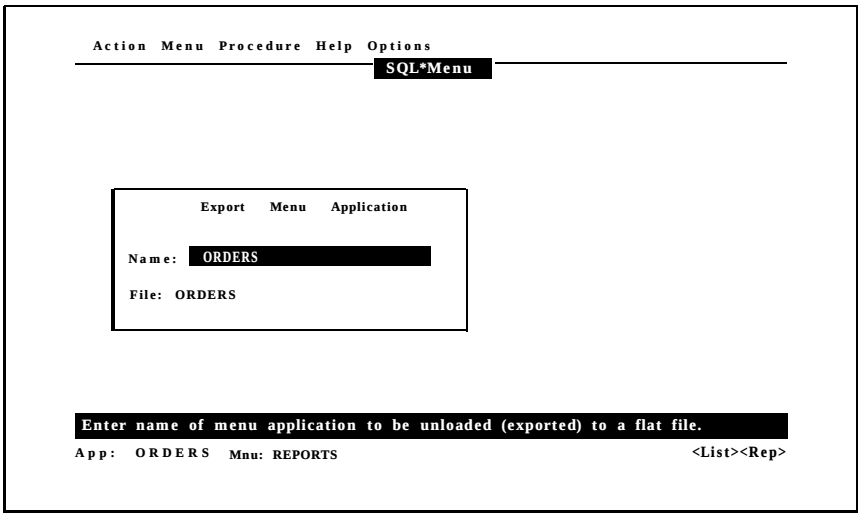
You can use the following method to move an application from one database to another. This is useful for backing up applications and for transporting them to different computer systems.

The following steps describe how to move a menu application:

1. Select the Action item from the main menu.
2. Select the Unload item from the Action submenu.

   The Export Menu Application dialog box appears (Figure 5-28) with the current application name in the Name field and its file name (specified in the Application Definition form) in the File field.

FIGURE 5-28
Export Menu Application Dialog Box



3. To export a menu application other than the current one, enter the name-of the application you want to export in the Name field of the Export Menu Application dialog box.

   To select an existing application name from a list of values, press [List] when the cursor is in the Name field.

4. To specify a new output file name for the application, move the cursor to the File field by pressing [Next Field] and enter the name.

   You can also enter a path prefix to specify the directory for the output file.

   The default directory is the current directory, from which you invoked SQL*Menu.

Note: Do not enter a file extension (such as `.SQL`) in the File field. Make sure that the file name satisfies the naming constraints of the system to which you will move the application.

5. Press [Accept], or press [Next Field] from the File field, to create an output file named *filename.SQL,* where *filename* is the value in the File field or the application's file name, if the File field is blank.

   SQL*Menu displays the `Working . . .` message while creating the output file, then returns to the main menu.

6. Transfer *filename.SQL* to the new location.

7. If the new location already has an application named *filename,* either rename or delete the existing application.

8. Enter SQL*Plus and type `START` *`filename`*. The username with which you enter SQL*Plus must be authorized as a SQL*Menu designer or DBA for the database into which you are importing the application.

   SQL*Plus imports the application into the database.

Note: Before you can run the imported application, you must generate a library file for it in the new location. Your SQL*Menu DBA must also create the roles that have access to the application and enroll users, because SQL*Menu does not export the information in the security table along with the application.

Security Operations

This section describes the security tasks for which the SQL*Menu DBA is responsible:

- granting access to SQL*Menu
- revoking access to SQL*Menu
- creating roles
- assigning users to roles

Adding anew user to SQL*Menu is a two-step process:

1. Grant the user access to SQL*Menu by enrolling the user's ORACLE username in SQL*Menu, with specific user privileges.

2. Assign the user to one or more roles.

   This step is optional for menu designers and DBAs, but role membership is necessary for ruining menu applications. New users cannot log in to SQL*Menu (Run Menu) unless they are members of at least one role that has access to a menu application.

Note: Access to a menu application is specified by the application designer on an item-by-item basis. The members of a role can run any menu application that contains items to which that role has access.

Granting Access to SQL*Menu

Granting a user access to SQL*Menu can be done by DBAs in the SQL*Menu (Design) interface, as described below, or by entering the command `GENMENU -gx` at your operating system's prompt, where *-gx* is one of the "grant" switches (`-g`, `-ge`, `-gd`, and `-ga`) described in "SQL*Menu System Components" in Chapter 12.

User Authorization

Every SQL*Menu user must have access to your computer's operating system. Once the user has this access, your ORACLE DBA can grant the user access to the ORACLE RDBMS by giving the user an ORACLE username and password. With these two access privileges, the SQL*Menu DBA can grant the user access to SQL*Menu.

SQL*Menu is installed with one user who has DBA privileges. This user normally has the username SYSTEM. For installation information, see the *ORACLE Installation and User's Guide* for your computer operating system.

User Privilege Levels

The SQL*Menu DBA can give a user one of the following levels of access privileges:

Application Operators These users can log into SQL*Menu (Run Menu) and run any existing application for which they have authorization. (Authorization to run an application is determined by role membership.) Application operators cannot create new applications or give others access to the system. They have SELECT privilege on the SQL*Menu base tables.

Application Designers These users also can log into SQL*Menu (Run Menu) and use existing applications for which they have authorization. In addition, application designers can use SQL*Menu (Design) to create new applications and generate documentation. Designers have SELECT, INSERT, UPDATE, and DELETE privileges on the SQL*Menu base tables.

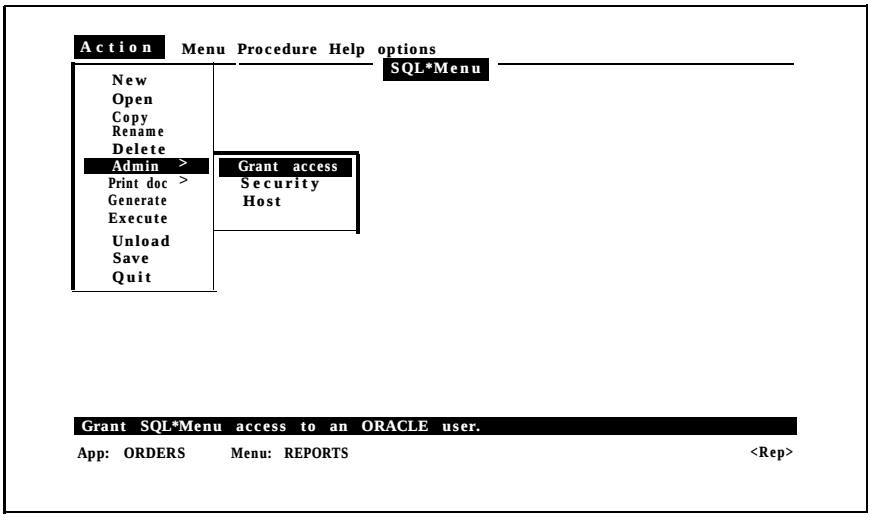
SQL*Menu DBA DBAs can control user access to SQL*Menu (Run Menu) and SQL*Menu (Design). Besides creating and documenting new applications, DBAs can access and modify any user-designed menu application. DBAs have GRANT privileges on the SQL*Menu base tables, as well as all the privileges that designers have (SELECT, INSERT, UPDATE, and DELETE).

The following steps describe how the SQL*Menu DBA can grant a user access to SQL*Menu:

1. In the SQL*Menu main menu, select the Action item.
2. In the Action submenu, select the Admin item.

   The Administration submenu appears (Figure 5-29).

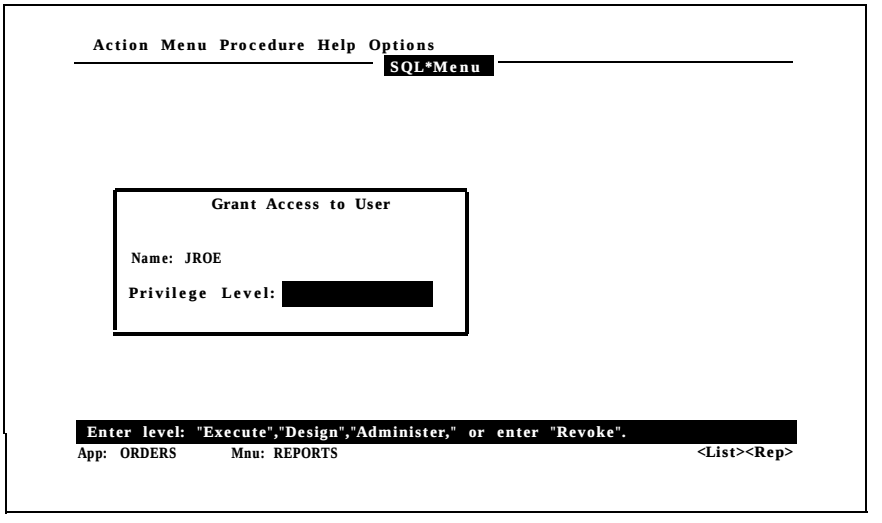
FIGURE 5-29
The Administration Submenu



3. Select the Grant Access item.

   The Grant Access to User dialog box appears (Figure 5-30).

FIGURE 5-30
Grant Access to User Dialog Box



4. Type the user's ORACLE username in the Username field and press [Next Field].
5. Enter the user's privilege level in the Privilege Level field:
   - `Execute` for application operators
   - `Design` for application designers
   - `Administrate` for SQL*Menu DBAs

   To select the privilege level from a list of values, press [List] when the cursor in the Privilege Level field.
6. Press [Accept] or [Next Field].

   SQL*Menu displays the `Working . . .` message while granting the access you specified, then returns to the main menu.

Revoking Access to SQL*Menu

The following steps describe how the SQL*Menu DBA can drop a user from SQL*Menu, revoking all privileges

1. In the main menu of the design interface, select the Action item.
2. In the Action submenu, select the Admin item.

   The Administration submenu appears (Figure 5-29).
3. Select the Grant Access item.

   The Grant Access to User dialog box appears (Figure 5-30).
4. Type the user's ORACLE username in the Username field and press [Next Field].
5. Enter `Revoke` in the privilege Level field.
6. Press [Accept] or [Next Field].

   SQL*Menu displays the `Working ...` message while revoking access privileges for the specified user, then returns to the main menu.

Creating Roles

As DBA, you define *roles* that enable users to run specific applications. For each role, you select the appropriate application options (background menu, operating system commands, and debug mode).

Once you have created the roles, application designers can grant them access to a menu application by using the Item Role window to specify access to individual items in the application. For information about how to use the Item Role window, see 'Defining a Menu Item" in this chapter.

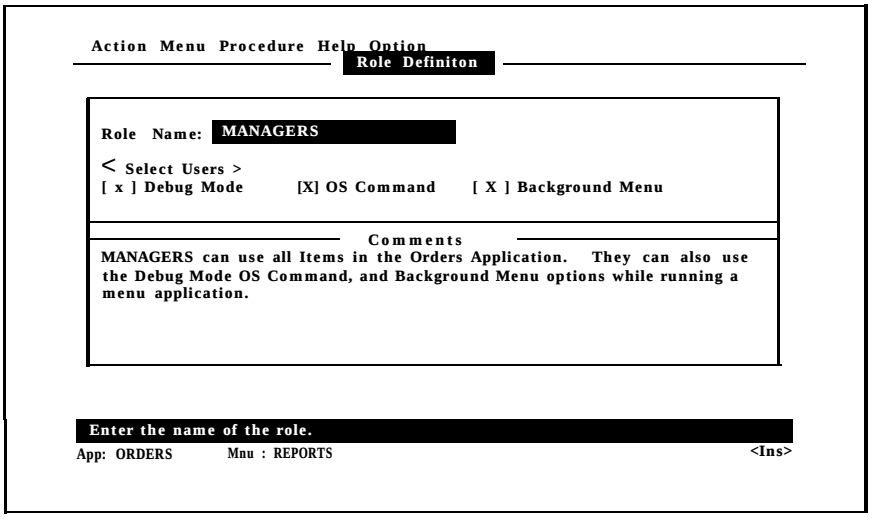
Note: When you install SQL*Menu Version 5.0, you can automatically upgrade applications created with SQL*Menu Version 4.1, converting the work-classes of Version 4.1 to roles. For information about upgrading applications, see Appendix A, "Upgrading SQL*Menu."

The following steps describe how the SQL*Menu DBA can create roles:

1. Select the Action item from the main menu of the design interface.
2. Select the Admin item from the Action submenu.
3. Select the Security item from the Administration submenu (Figure 5-29).

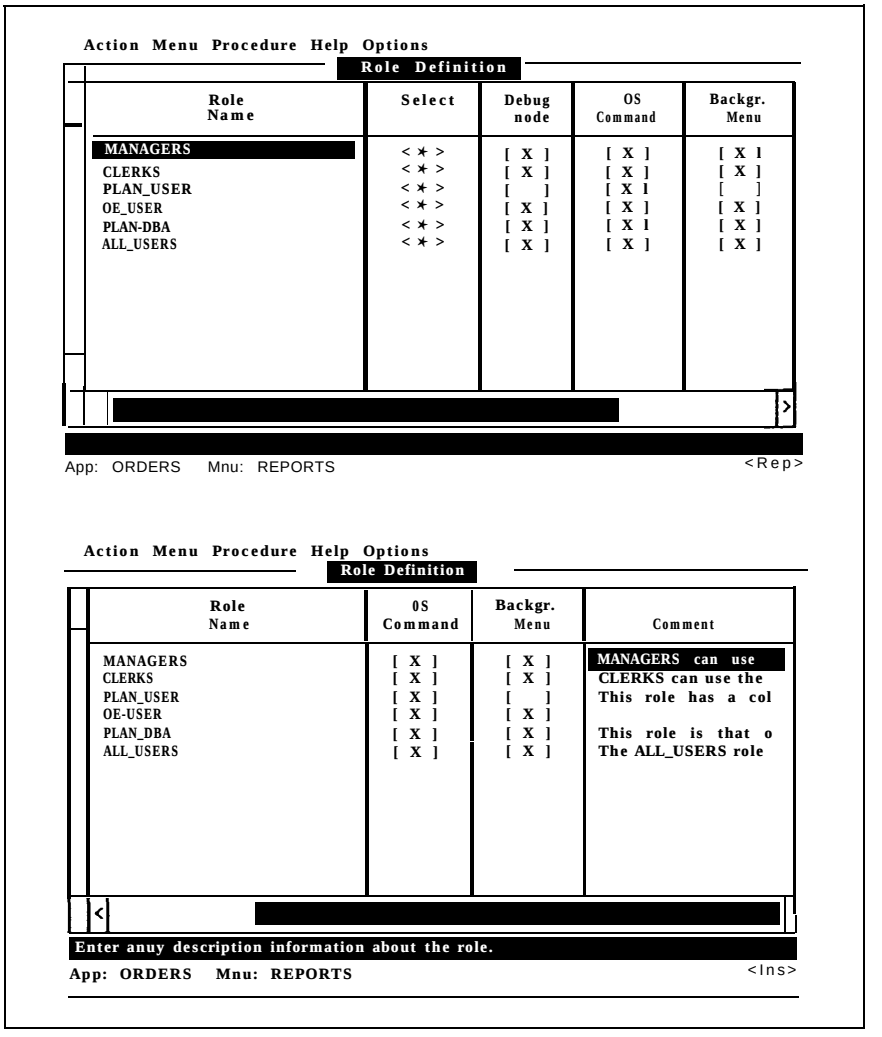
   The Role Definition form or spread table appears (Figures 5-31 and 5-32). To switch between the form and the spread table, press [Change Display Type].

FIGURE 5-31
Role Definition Form



To scroll the spread table horizontally, press [Next Field] or [Previous field]. Figure 5-32 shows the Role Definition spread table as it appears when first displayed (top) and when the cursor is in the last column (bottom).

FIGURE 5-32
Role Definition Spread Table



4. Press [Insert Record] to create a blank record and type the name of a role in the Role Name field.
5. Press [Next Field] twice to move the cursor past the Select Users button to the Debug Mode check box. To give the role the Debug Mode option, press [Select] or type any nonblank character.

   For information about the Debug Mode option, see "Executing a Menu Application" earlier in this chapter.
6. Press [Next Field] to move the cursor to the OS Command check box. To give the role the OS Command option, press [Select] or type any nonblank character.

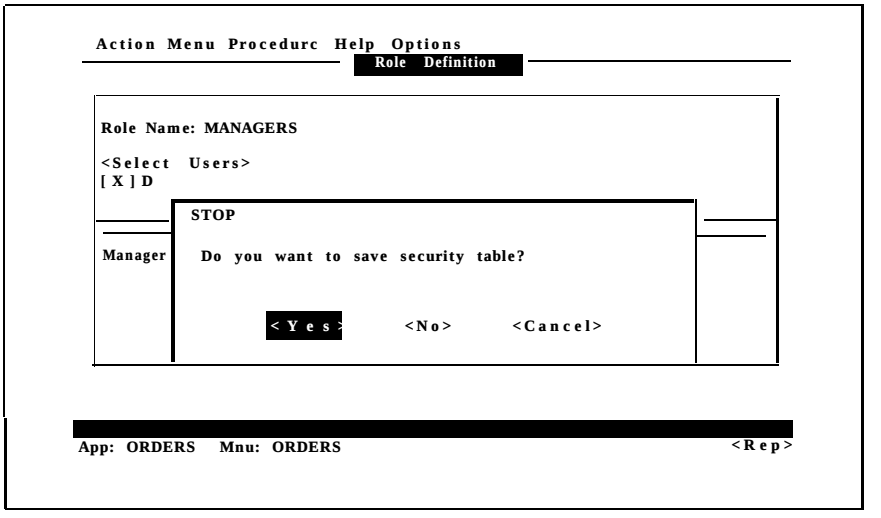
   For information about the OS Command option, see "Using Operating System Commands" in Chapter 2.
7. Press [Next Field] to move the cursor to the Background Menu check box. To give the role the Background Menu option, press [Select] or type any nonblank character.

   For information about the Background Menu option, see "Using the Background Menu" in Chapter 2.
8. Press [Next Field] and type a description of the role in the Comments scroll region.
9. Repeat steps 4 through 8 to create another role. To delete a role, press [Down] or [Up] to move the cursor to the record for that role, then press [Delete Record].
10. To assign individual users to the newly created roles, use the Select Users button as described in the following section, "Assigning Users to Roles."
11. Press [Accept] to leave the Role Definition form or spread table.

    An alert appears with the message `Do you want to save security table?` and three buttons for your response (Figure 5-33):

    - `( Yes )` saves new role definitions in the security table.
    - `( No )` exits without saving your changes.
    - `( Cancel )` returns to Role Definition without saving.

FIGURE 5-33
Alert for Saving Role Definitions



12. Press [Select] to save the new information in the security table, or press [Next Field] to move the cursor to a different response before pressing [Select].

    If you select (Yes) or (NO) , SQL*Menu returns to the main menu of the design interface. If you select (Cancel) , SQL*Menu returns to the Role Definition form or spread table.

Assigning Users to Roles

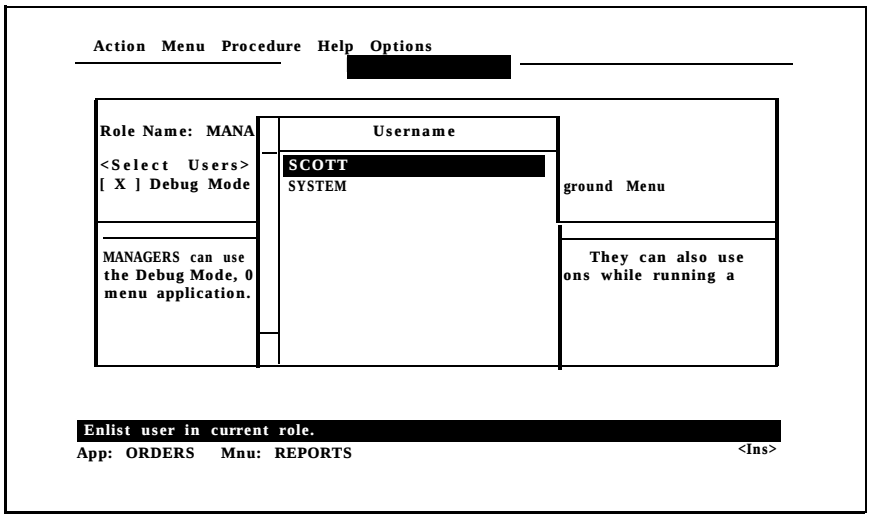
One of the responsibilities of the SQL*Menu DBA is to assign users to roles so that they can run menu applications.

The following steps describe how the SQL*Menu DBA can assign users to a role:

1. Select the Action item from the main menu of the design interface.
2. Select the Admin item from the Action submenu.
3. Select the Security item from the Administration submenu.

   The Role Definition form or spread table appears. To switch between the form and spread table, press [Change Display Type].
4. In the Role Name field, press [Down] to move the cursor to the name of the role in which you want to enroll one or more users.
5. Press [Next Field] to move the cursor to the Select Users button, then press [Select] to display the Username window (Figure 5-34).

FIGURE 5-34
The Username Window



6. Press [Insert Record] and type an ORACLE username.
7. Repeat step 6 to enroll more users in the current role.
8. To delete a username, press [Down] or [Up] to move the cursor to that username, then press [Delete Record].
9. Press [Accept] to return to the Role Definition form or spread table.
10. To enroll users in another role, repeat steps 4 through 9.
11. Press [Accept] to leave the Role Definition form or spread table.

    An alert appears with the message `Do you want to save security table?` and three buttons for your response

    - `( Yes )` saves new role definitions in the security table. .
    - `( NO )` exits without saving your changes.
    - `( Cancel )` returns to Role Definition without saving.
12. Press [Select] to save the new information in the security table, or press [Next Field] to move the cursor to a different response before pressing [Select].

    If you select `( Yes )` or `( No )` , SQL*Menu returns to the main menu of the design interface. If you select `( Cancel )` , SQL*Menu returns to the Role Definition form or spread table.

CHAPTER

6

ELEMENTS IN THE DESIGN INTERFACE

This chapter describes the interface elements that you use to design a menu application in SQL*Menu, and explains how to use them. The chapter covers the following topics:

- context in the design interface
- menu elements
- data-entry elements
- the online help system

Before you read this chapter, you should become familiar with the SQL*Menu concepts introduced in Parts I and II.

Note: This chapter describes interface elements as they appear in a character-mode environment. These elements might appear differently in other environments. In addition, how you interact with these elements might be different in other environments.

Context in the Design Interface

In SQL*Menu, *context* determines what actions you can perform in various parts of the interface, and what objects your actions affect. The *current context* is your current position on an element in the interface and the current selection of objects within the object hierarchy.

To modify an object definition, that object must be part of the current context.

Hierarchy and Context

SQL*Menu establishes context from the top of the hierarchy down. That is, SQL*Menu must have an application context before it can establish a menu context, and it must have a menu context before it can establish an item context.

For example, when you enter the Item Definition spread table, your current context includes the application, a menu, and an item belonging to that menu. The current context of interface elements includes the spread table, the current record, and the current field (or other interface element).

The status line shows the current application and menu. When you leave an object definition form or spread table, the established object context remains the same until you select another object.

Selecting an object affects the context for all objects lower in the hierarchy. For example, if you return to the main menu from the Menu Definition form, the status line continues to show the name of the menu most recently selected. When you enter the Menu Definition form again, the cursor appears on the record defining that menu; and if you enter the Item Definition Form, you only see the items for that menu. Selecting a different item does not affect the menu context, but selecting a different menu changes the items that you can select.

Function Keys and Context

Many function keys are context-sensitive, either because they act differently indifferent elements of the interface or because they act upon the objects of the current context.

Some function keys allow you to change the object context, such as [Up] and [Down], [Navigate], and [Insert Record].

[Zoom Out] and [Zoom In] move you up and down the object hierarchy, respectively, within the object context.

For descriptions of these function keys, see Chapter 7, “Function Keys for Designers.”

Menu Elements

The menu elements in SQL*Menu (Design) are similar to those in SQL*Menu (Run Menu), except for differences in the status line. Only the pull-down menu style is available in SQL*Menu (Design).

For general descriptions of the menu elements in SQL*Menu (Run Menu), see Chapter 3, "Menu Styles and Navigation."

The menu display in SQL*Menu (Design) can include these elements:

- menu and submenu
- menu items
- message line
- status line

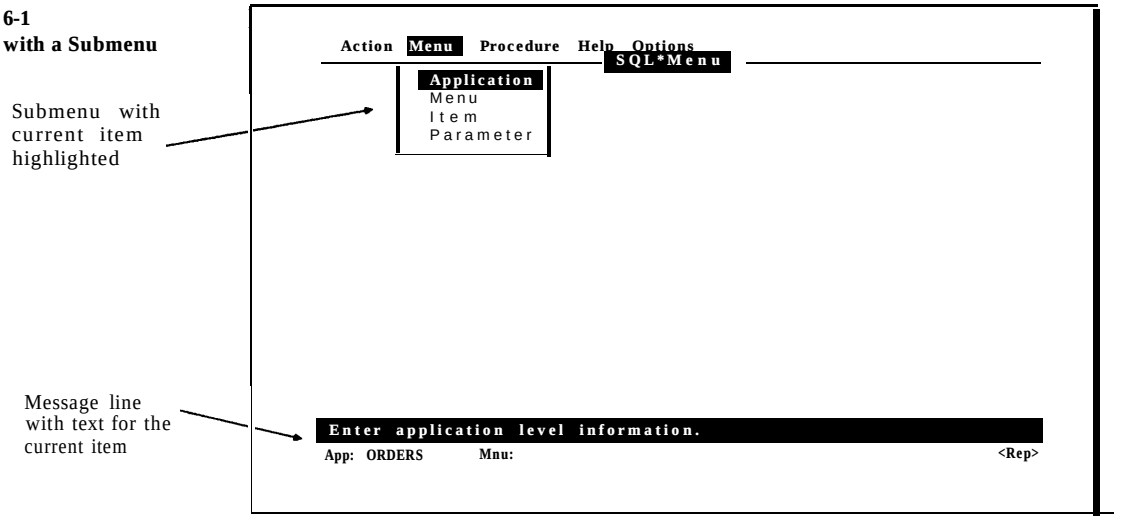
Menu

A *menu* is a list of choices from which you select your next action in SQL*Menu, which can be going to another menu or executing a command.

A *submenu* is a menu called by an item on another menu, called its parent menu.

Figure 6-1 shows a sample menu, with a submenu, in SQL*Menu (Design). The current item is highlighted in the submenu, and the message line shows a hint for the current item.

FIGURE 6-1
A Menu with a Submenu



Menu Item

A *menu item* is a choice that you can select to perform an action, such as calling another menu or quitting an application.

You can select a menu item in several different ways:

- Use the function keys [Up], [Down], [Left], and [Right] to position the cursor on the menu item, then press [Select].
- In a bit-mapped environment, select the menu item with a mouse.
- Type the first capital letter of the item name, if that item is the only one with that first capital letter (or is the next item with that initial, relative to the cursor).

Unavailable items are listed in grey characters, rather than bold. What items are available in SQL*Menu (Design) depends on context; for example, you can't define a menu item until you select the menu.

Note: The roles that determine item availability when you are ruining an application with SQL*Menu (Run Menu) do not have any effect in the design interface.

Menu items in the design interface appear in a horizontal menu bar for the main menu, and items in submenus appear in the pull-down display style.

Message Line

The *message line,* or *hint line,* is the next-to-last line on the menu screen, just above the status line. The message line displays instructions, hints, warnings, and error messages.

The Suppress Hints Option

If you do not need the messages during a design session, you can select the Options item from the main menu and then select the Suppress Hints option in the Option Selection form, as described in "Setting Options in SQL*Menu (Design)" in Chapter 5.

The message line still displays warnings and error messages when the Suppress Hints option is on.

Status Line

The *status line* is the bottom line on the menu screen, just below the message line. It provides information about the current context and option settings.

The status line for SQL*Menu (Design) gives the following information:

`App: application_name`	Gives the name of the current SQL*Menu application. Displays the first ten characters.
`Mnu: menu_name`	Gives the name of the current menu. Displays the first ten characters.
`<List>`	Indicates that there is a list of values for the current field. If there is no list of values for the current field this area of the status line appears blank.
`<Rep>` or `<Ins>`	Indicates the typing mode, Replace or Insert characters.

Data-Entry Elements

This section describes the interface elements that you use for entering data and selecting actions while designing a menu application with SQL*Menu. The interface elements are described in alphabetical order

- alert
- button or radio button
- check box
- dialog box
- field
- form
- list of values
- scroll region
- spread table

Alert

An *alert* is a window that notifies you of a condition that has arisen due to your last action, or that will arise unless you take a preventive action. It forces you to respond to the condition by either acknowledging the alert or selecting an appropriate action.

An alert contains a label, a text message, and one or more action items. The label indicates the severity of the alert:

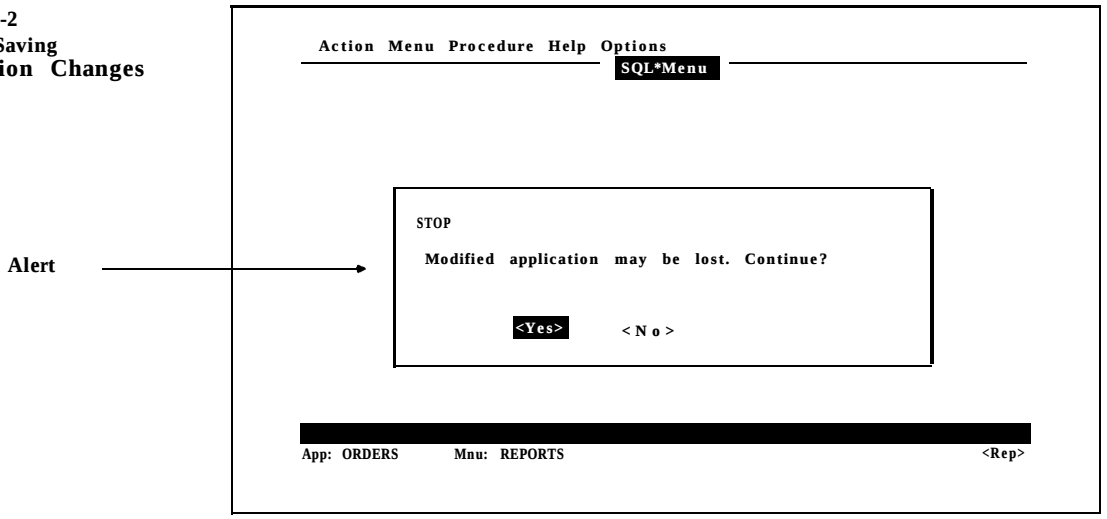
`CAUTION`	The alert contains a cautionary statement or question. You must confirm or cancel your action.
`NOTE`	The alert contains an informational statement- You must acknowledge the alert.
`STOP`	The alert identifies a severe problem and may require acknowledgement or corrective action.

The text message explains the current situation that activated the alert. The action items let you determine what action SQL*Menu should take or acknowledge the current condition. When you select an action, the alert disappears and SQL*Menu executes the appropriate action.

To select an action item in an alert, press [Right] or [Left] to move the cursor to the item you want, then press [Select] or Return. You can also press [Next Field] or [Previous Field] to move the cursor.

Figure 6-2 shows an alert for saving a modified application. This alert appears when you try to quit a menu application, or open a different one, without saving modifications to the current application.

FIGURE 6-2
Alert for Saving Application Changes



Button or Radio Button

A *button* appears as a text label in parentheses. The label can be an asterisk, `( * )`, or other symbol, such as `( Select Users )`. When you select a button by pressing [Select], SQL*Menu displays a window or a dialog box, or performs an action such as setting an option.

- If a list appears, you can enter values into the list or select one or more values from the list. When you press [Accept], SQL*Menu saves your choices and the list disappears.
- If a dialog box appears, you can assign values in the fields and select options in the check boxes. When you press [Accept], SQL*Menu saves your choices and the dialog box disappears.
- If the button is a *radio button,* pressing [Select] switches to the next option setting in the set of radio buttons.

Radio buttons are labeled by (o) in the Option Selection form. Only one of the radio buttons can be selected at a time. The cursor only moves to the current selection in the set of radio buttons, except when you press [Select] to change the option setting.

Figure 6-3 shows the Select Users button in the Role Definition form, and Figure 6-4 shows the radio buttons in the SQL*Menu (Design) Option Selection form.

FIGURE 6-3
A Button in the Role Definition Form

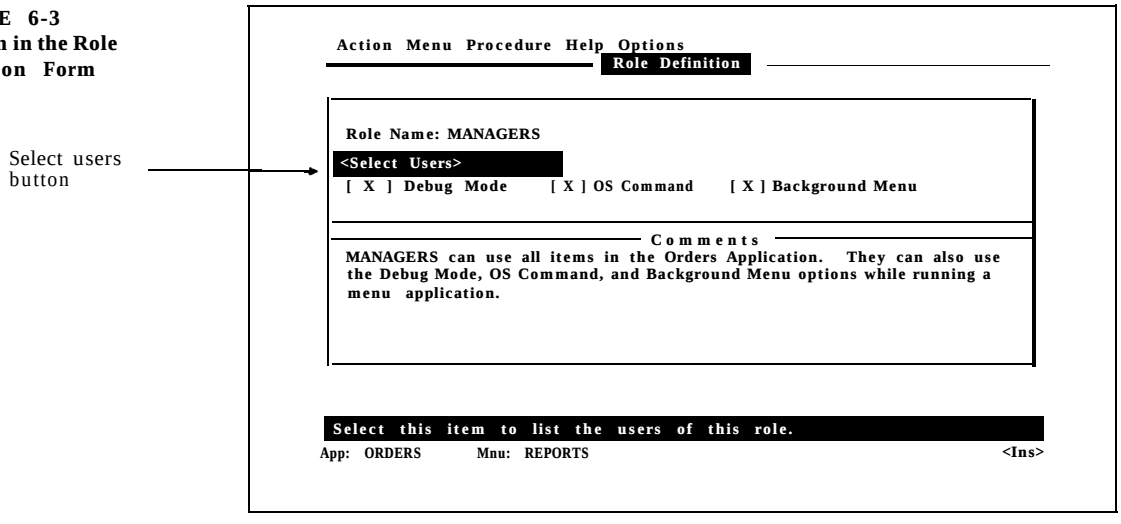
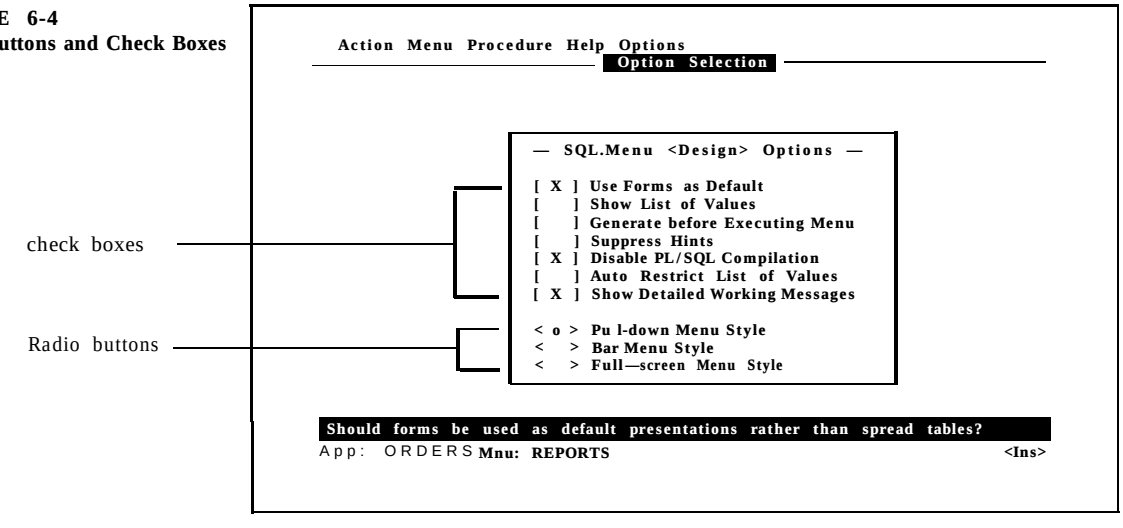


FIGURE 6-4
Radio Buttons and Check Boxes



Check Box

A *check box* is a two-choice list from which you select one choice: "on" or "off."

- When a check box contains a blank space, the characteristic associated with the check box is turned off.
- When a check box contains an "X," the characteristic associated with the check box is turned on.

Press [Select] to toggle (turn on or off) the characteristic associated with a check box. Alternatively, type any character to turn the characteristic on or type a blank space to turn it off.

Figure 6-4 shows the check boxes in the SQL*Menu (Design) Option Selection form.

Dialog Box

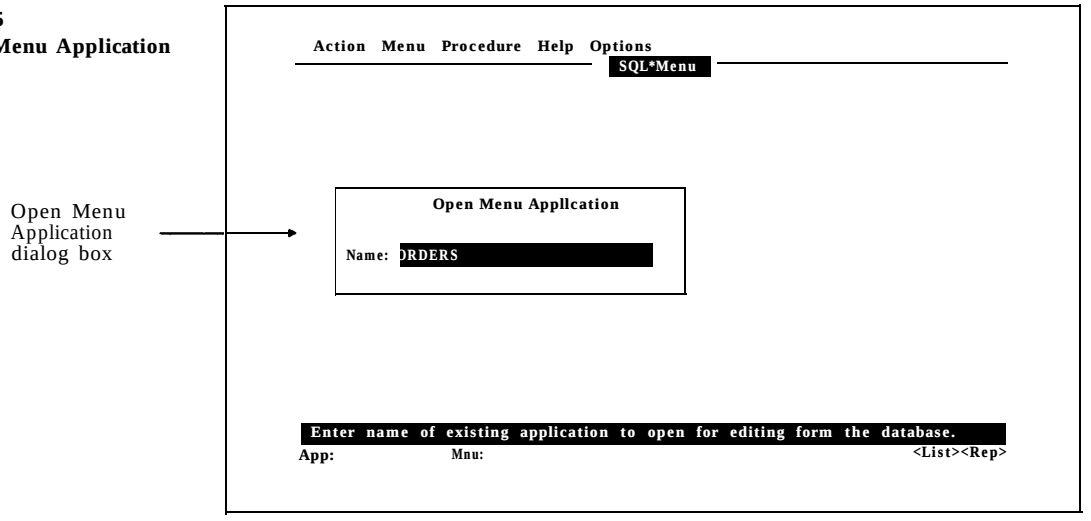
A *dialog box* is a small form that overlies a portion of the current display. It contains one or more fields which you can fill in or edit.

The information you enter in a dialog box can determine what object SQL*Menu should act on. For example, you enter the name of a menu application in the Open Menu Application dialog box.

When you press [Accept], the dialog box disappears and SQL*Menu takes the appropriate action.

Figure 6-5 shows the Open Menu Application dialog box.

FIGURE 6-5
The Open Menu Application Dialog Box



Field

A *field* in the design interface is an area where you can enter, edit, or delete data. That data either defines an object, as with the Menu Name field in the Menu Definition form, or assigns a value to a variable.

Fields appear in forms, spread tables, and dialog boxes.

Figures 6-6 and 6-7 show the Menu Name field as it appears in the Menu Definition form and spread table, respectively.

FIGURE 6-6
A Field in the Menu Definition Form

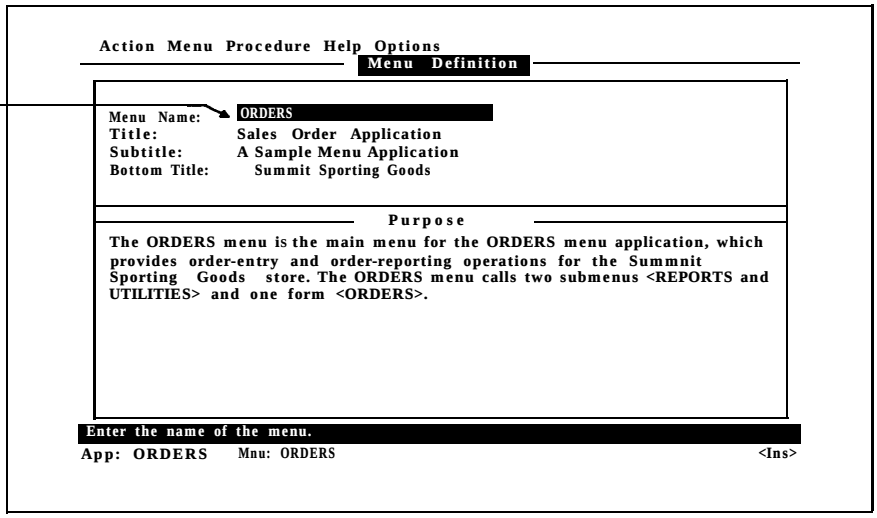
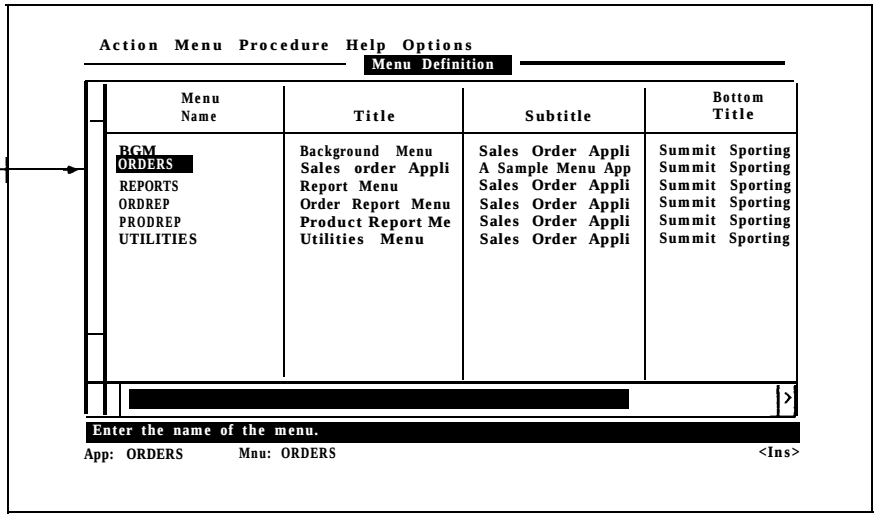


FIGURE 6-7
A Field in Menu Definition Spread Table



Field Validation SQL*Menu validates field entries based on other information that you've already specified for the application. If a field fails validation, an error message prompts you for corrective action.

Form

A *form* in the design interface is an area that can contain fields, check boxes, buttons, and scroll regions. You navigate to the elements in a form by pressing [Next Field].

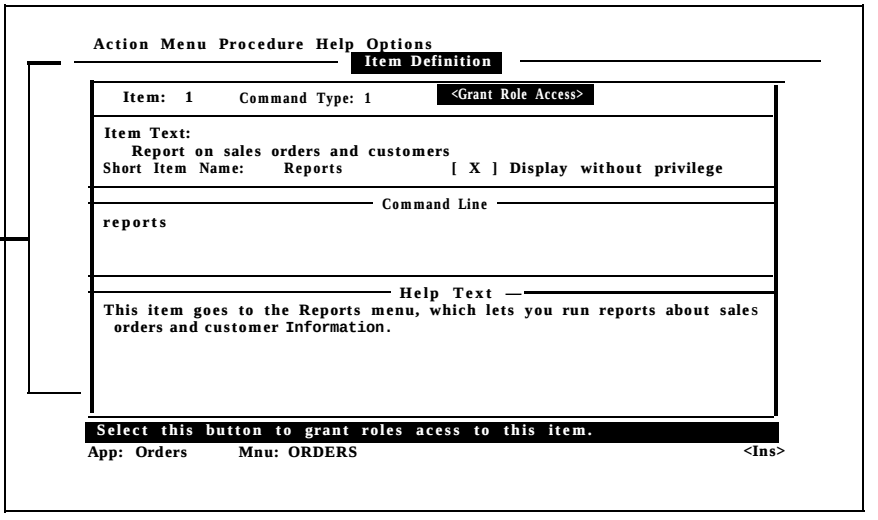
A form shows one record of information that only applies to one object.

You can add information for anew object by pressing [Insert Record], or delete an object by pressing [Delete Record].

When you press [Accept], SQL*Menu temporarily saves the values or settings in the form and returns to the main menu.

Figure 6-8 shows the Item Definition form.

FIGURE 6-8
The Item Definition Form



Form Context Forms always display a record that is in context with the current object.

- The Menu Definition form displays a record for a menu that belongs to the current application.
- The Item Definition form displays a record for an item that belongs to the current menu.

You can move to another record by pressing [Next Record] or [Previous Record] in any form element, or by pressing [Down] or [Up] in any form element except a scroll region.

In the design interface most object definition forms have corresponding spread tables, but there is no spread table for the Application Definition form. A form display is most useful for defining one object at a time, when you do not need to see the other object definitions.

Press [Change Display Type] to switch between form displays and spread table displays. You can also change the default display type from spread table to form by selecting the Use Forms as Default option in the Option Selection form, as described in "Setting Options in SQL*Menu (Design)" in Chapter 5.

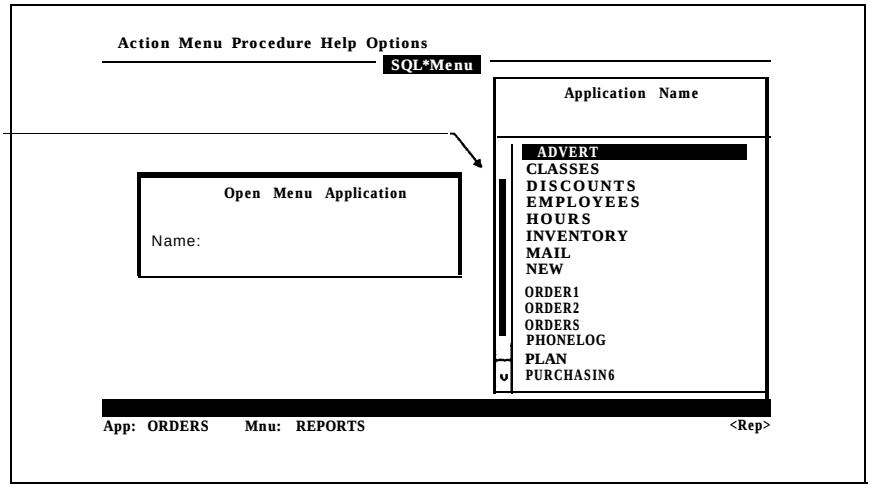
List of Values

A *list of values* is a window that displays the existing values for the current data-entry field in the interface. You can use a list of values to select a value or to search for all the values that match a search string.

A list of values has a title, a list area, and usually a Find field. You can select a value in the list area, or you can return to the data-entry field by pressing [Cancel].

Figure 6-9 shows a list of values for the Name field in the Open Menu Application dialog box.

FIGURE 6-9
List of Values
for the Name Field



Note: Some fields only let you pick existing values, but others can take any valid entry. For example, you can only use an existing value if you are naming an object to copy, but you can use any valid name if you are creating a new object.

A vertical scroll bar appears at the left side of the list of values. Arrows in the scroll bar indicate if the additional values are “above” or “below” the displayed screen.

A vertical elevator indicates the relative vertical position of the current screen display in the list of values. The size of the elevator indicates the proportion of values that are currently displayed.

List Area The *list area* in a list of values is a modified scroll region containing one or more values, with a scroll bar at the left. When you enter the list area, the first value is highlighted to show that it is the current selection. To pick another value as the current selection, use [Up] and [Down] (or a mouse, in a bit-mapped environment) to move the cursor to that list item.

To select a value in the list area, make that value the current selection (highlighted) and press [Select]. SQL*Menu automatically enters your choice in the field.

List of Values Search Function The Find field in a list of values lets you display a sublist of all values that match a specified search pattern. To move the cursor from the list area to the Find field (or back to the list area), press [Next Field].

To search for a range of values, type a search pattern in the Find field and press [Next Field] or [List]. If any values match the pattern, SQL*Menu automatically reduces the list to the matching values and highlights the first value. If no values match the pattern, the cursor returns to an empty list area.

Note: The search function is case-insensitive.

The search pattern can include the standard SQL wildcards:

%	This wildcard represents any number of characters, anywhere in the pattern. For example, “%S% matches every value that contains at least one letter S.
_(underscore)	This wildcard represents any one character. For example, “_REA” matches the values AREA, BREA, CREA, and so on, but not the value REA.

The % wildcard is implicit at the end of every search pattern. For example, “%S” is the same as “%S%”, and “_REA" is the same as “_REA%”.

You can search through a list of values repeatedly, with increasingly specific search patterns, by returning to the Find field and changing the search pattern. To delete parts of the search pattern, position the cursor and use [Delete Character]. You can also redisplay the full list of values. by searching for an empty string or "%".

The Show List of Values Option

If you check the Show List of Values check box in the Option Selection form, a list of values appears automatically whenever one is available for the current field. Otherwise, if `<List>` appears in the status line you can press [List] to display a list of values for the current field.

The Auto Restrict List of Values Option

If you check the Auto Restrict List of Values check box in the Option Selection form, SQL*Menu automatically restricts a list of values to values that match the search pattern in the corresponding field.

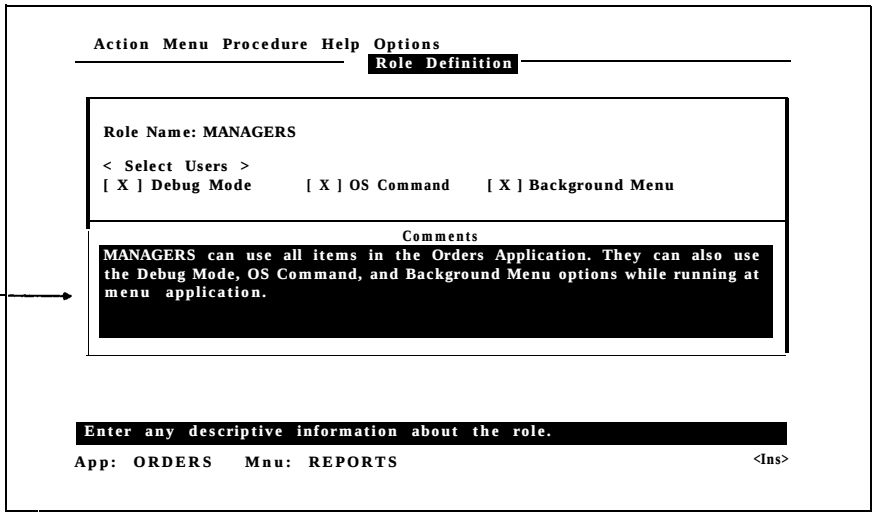
For more information about the Show List of Values option and the Auto Restrict List of Values option, see "Setting Options in SQL*Menu (Design)" in Chapter 5.

Scroll Region

A *scroll region* consists of an area that allows you to scroll vertically through several lines of information.

Figure 6-10 shows the Comments scroll region in the Role Definition form.

FIGURE 6-10
Scroll Region in the Role Definition Form



In a scroll region, as in a field, you can enter, edit, and delete data. You can move through a scroll region with editing function keys such as [Search], which invokes the scroll region search function.

Scroll Region Search Function The scroll region search function allows you to look for a specific value in a scroll region. To do so, press [Search] to call up the Search dialog box.

In the Search dialog box, specify the following search criteria:

- the text string (case-insensitive) for which you want to search
- an optional text string with which to replace the search string
- whether to search/replace forward or backwards from the current cursor position
- whether or not to replace all found instances of the search string with the replace string

Spread Table

A *spread table* is a screen display that can span more than one screen, both horizontally and vertically. It can contain fields, check boxes, buttons, scroll regions, and horizontal and vertical scroll bars.

The names or titles of the fields check boxes, buttons, and scroll regions are arranged horizontally across the top of the spread table. The actual elements are arranged in vertical columns below the titles.

Information in a spread table that applies to a single object appears in one record. A record in a spread table is a horizontal row across all columns.

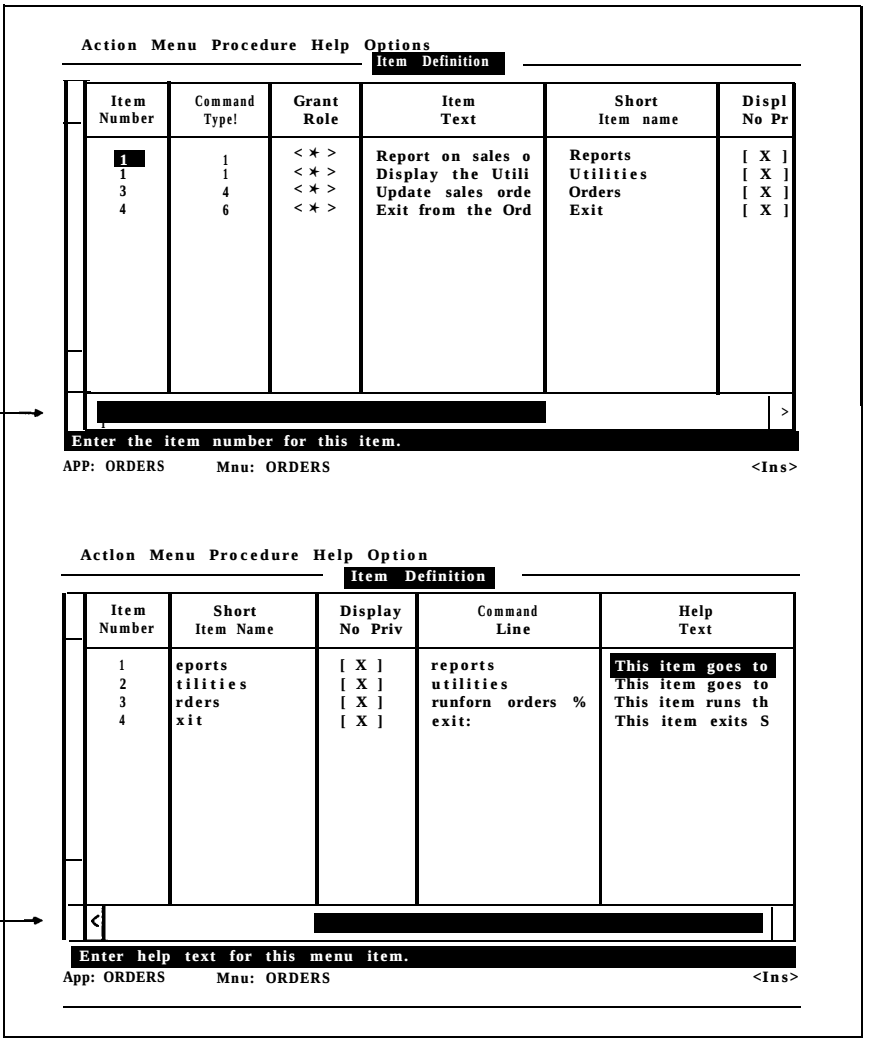
You can add a new record by pressing [Insert Record], or delete a record by pressing [Delete Record].

Scroll regions in a spread table only display one line of data at a time. To change lines, press [Down] or [Up] while the cursor is in the scroll region. To display several lines at once, switch to the form display by pressing [Change Display Type].

You navigate to the elements in a spread table by pressing [Next Field]. When you press [Accept], SQL*Menu temporarily saves the values and settings in the spread table and returns to the main menu.

Figure 6-11 shows the Item Definition spread table as it appears when first displayed (top) and when the cursor is in the last column (bottom).

FIGURE 6-11
Item Definition Spread Table



Spread Table Context Spread tables always display records that are in context with the current object.

- Spread tables associated with menus display all records that apply to the current menu application.
- Spread tables associated with menu items display all records that apply to the current menu.

You can move to another record by pressing [Next Record] or [Previous Record] in any spread table element, or by pressing [Down] or [Up] in any spread table element except a scroll region.

In the design interface every spread table is associated with a form. A spread table display is generally most useful for navigating to a particular object and reviewing the contents of an application.

Press [Change Display Type] to switch between spread table displays and form displays. The default display type is the spread table, unless you specify forms for the default by selecting the Use Forms as Default option in the Option Selection form, as described in "Setting Options in SQL*Menu (Design)" in Chapter 5.

Spread Table Scrolling A horizontal scroll bar appears at the bottom of a spread table. Arrows in the scroll bar indicate if the spread table has additional fields "before" or "after" the displayed screen.

A bar graphic, known as an *elevator,* indicates the relative horizontal position of the current screen display in the spread table. The size of the elevator indicates the proportion of the spread table that is currently displayed.

If a spread table spans more than one screen, it scrolls horizontally one screen at a time. (In a bit-mapped environment, the scrolling is continuous, not by screen.) Spread tables always wrap from the last column of the spread table to the first column of the spread table and from the first column to the last column.

A vertical scroll bar appears at the left side of the spread table if the spread table contains more records than it can display on one screen. Arrows in the scroll bar indicate if the additional records are "above" or "below" the displayed screen.

A vertical elevator indicates the relative vertical position of the current screen display in the spread table. The size of the elevator indicates the proportion of records that are currently displayed.

The Online Help System

The online help system contains information about elements in the design interface, function keys, and the SQL and PL/SQL languages. You can invoke the online help system by two methods:

- Press the [Help] function key from any interface element in SQL*Menu (Design).

  SQL*Menu displays context-sensitive information and gives you access to the whole help system.

- Select the Help item in the SQL*Menu (Design) main menu, then select the Help system item in the Help submenu.

  SQL*Menu displays the introduction, “About the Online Help System," and gives you access to the whole help system.

To exit the online help system, either press [Cancel] or press [Menu] and select the Exit item.

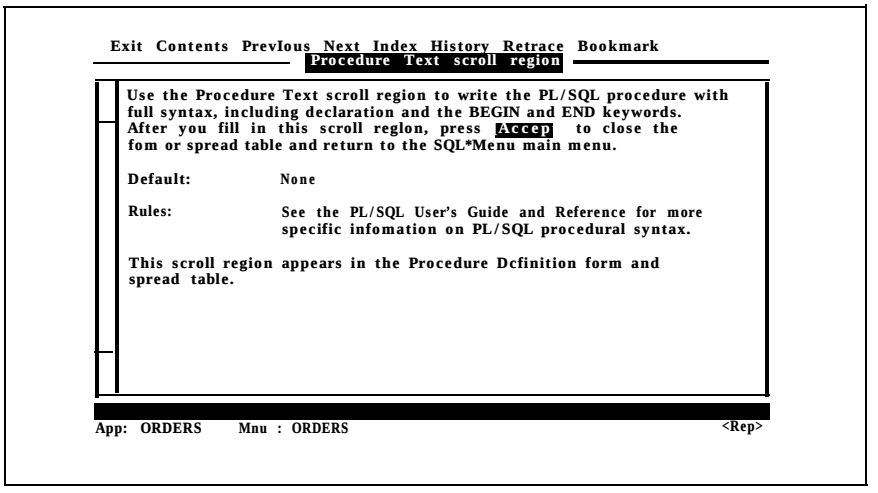
Displaying Context-Sensitive Help

You can display context-sensitive help in the design interface by pressing [Help] from any interface element.

When the message line contains an error message, pressing [Help] displays help for the error message rather than an interface element.

Figure 6-12 shows the help information for the Procedure Text scroll region of the Procedure Definition form or spread table.

FIGURE 6-12
Help for Procedure Text Scroll Region



Navigating in the Online Help System

Within the online help system, you can either scroll to additional screens of help text, display help for a highlighted key word, or press [Menu] to move to the help system menu. In the menu, you can select items that allow you to navigate through the help system by several methods.

The introductory text, About the Online Help System, explains the various methods of navigation. To display the first screen of About the Online Help System, use one of the following methods.

- Select the Help system item in the Help submenu of the design interface, as described in "Getting Help Information" in Chapter 5.
- Press [Help] within the online help system.
- Select the topic "About the Online Help System" in the Contents or Index.

An elevator in the vertical scroll bar at the left of the screen indicates the proportion of the available text that is currently displayed and your relative position within the text. Press [Scroll Down] and [Scroll Up] to scroll to another screen of help information.

Using Keyword Topics

To display help about a highlighted topic in the help text, move the cursor to the topic by pressing [Right] and [Left], or [Next Field] and [Previous Field], then press [Select].

Using the Table of Contents

The table of contents of the online help system lists the major help topics, including specific interface elements, function keys, and the SQL and PL/SQL languages.

The following steps describe how use the table of contents to navigate in the online help system:

1. Within the online help system, press [Menu].

   The cursor moves to the Exit item in the menu.

2. Press C to display the table of contents of the online help system.
3. Press [Scroll Down] or [Scroll Up] to scroll through the table of contents, and press [Down], [Up], [Left], or [Right] to move the cursor to a keyword topic.
4. Press [Select] to display help about the current topic.

Using the Index

The Index of the online help system contains a complete list of help topics arranged alphabetically.

The following steps describe how use the index to navigate in the online help system

1. Within the online help system, press [Menu].

   The cursor moves to the Exit item in the menu.

2. Press I to display the index.
3. Press [Scroll Down] or [Scroll Up] to scroll through the index, and press [Down], [Up], [Left], or [Right] to move the cursor to a keyword topic.
4. Press [Select] to display help about the current topic.
5. To display help about the next index topic, press [Menu] and select the Next item from the menu. To display help about the previous index topic, press [Menu] and select the Previous item.

Using the Bookmark

To mark your place in the online help system, either press [Bookmark] or press [Menu] and select the Bookmark item. You can then return to that place from outside the help system by pressing [Bookmark].

Using History and Retrace

To display a list of all the help topics you have read within the online help system, press [Menu] and select the History item. You can return to any topic in the list by moving the cursor to it and pressing [Select].

To retrace your steps through the topics of the online help system, press [Menu] and select the Retrace item.

CHAPTER

7

FUNCTION KEYS FOR DESIGNERS

This chapter describes the function keys that you can use in SQL*Menu (Design) and groups them into categories for the designer's interface. The function key groupings include the following categories:

- cursor movement functions
- editing functions
- general functions
- user assistance functions

Function Keys for SQL*Menu (Design)

The SQL*Menu (Design) functions can move the cursor around a screen, navigate to another screen, or execute an action. You perform these functions by pressing function keys. (In a bit-mapped environment, you can perform some functions with either the mouse or a function key.)

This book refers to function keys by their function names, rather than physical key names, because the physical keys that correspond to the functions are different for different computer systems. Function names always appear in square brackets, such as [Left] and [Right].

Many function keys for SQL*Menu (Design) are the same as function keys for SQL*Forms (Design), but some functions are unique to either SQL*Menu or SQL*Forms. For information about SQL*Forms (Design) function keys, see the *SQL*Forms Designer's Reference.*

Function Key Mappings

A *keyboard map* shows what physical keys correspond to the SQL*Menu functions. SQL*Menu has two keyboard maps, one for SQL*Menu (Design) and one for SQL*Menu (Run Menu).

While you are using SQL*Menu, you can display your terminal's keyboard map by pressing [Show Keys]. The resulting display lists the functions and the physical key names for the current context of SQL*Menu (Design).

The login screen shows you what key or combination of keys to press for [Show Keys] on your keyboard. [Show Keys] works within any context of the SQL*Menu (Design) interfaces, including dialog boxes, forms, and the online help system.

Function Key Descriptions

This section describes all of the functions available in SQL*Menu (Design). The function keys and their descriptions are listed alphabetically, and the following section groups them by related functions.

Note: The function key descriptions in this section are based on the default function key assignments. Some of these functions may not be available to you. Ask your DBA for a list of the available functions.

The following function keys are available in SQL*Menu (Design):

[Accept]*	[First Line]	[Print]*
[Beginnig of Line]*	[Help]*	[Quit]*
[Bookmark]	[Insert Record]	[Refresh]*
[Cancel]*	[Insert/Replace]*	[Right]*
[Change Display Type]	[Last Line]	[Scroll Down]
[Clear Field]*	[Left]*	[Scroll Left]
[Copy]	[List]	[Scroll Right]
[Copy Object]	[Menu]	[Scroll Up]
[Cut]	[Navigate]	[Search]
[Delete Backwards]*	[Next Field)*	[Select]*
[Delete Character]*	[Next Record]	[Show Keys]*
[Delete Line]	[Paste]	[Up]*
[Delete Record]	[Previous Field]*	[Zoom In]
[Down]*	[Previous Record]	[Zoom Out]
[End of Line]*		

* These function keys are also used in SQL*Menu (Run Menu).

The specific function that a function key performs can vary depending on the context in which you use it. The descriptions in this section provide context-sensitive information where applicable.

[Accept]

[Accept] terminates data input. It has no function in a menu.

In a dialog box [Accept] closes the dialog box and acts upon your entry.

In a form or spread table [Accept] closes the current form or spread table and preserves any changes you made.

[Beginning of Line]

[Beginning of Line] has no function in a menu.

In a field [Beginning of Line] moves the cursor to the first column (displayed or undisplayed) of the current field.

In a scroll region [Beginning of Line] moves the cursor to the first column of the current line.

[Bookmark]

In the help system [Bookmark] puts a placeholder on the current help screen.

In the design interface [Bookmark] displays the help screen on which you last put a placeholder.

[Cancel]

[Cancel] terminates the current operation and exits the current context, discarding any changes or selections. [Cancel] has no function in an alert.

In a menu [Cancel] redisplays the previous menu.

In a form, spread table, or dialog box [cancel]:

- cancels the current operation or action (such as opening a form)
- reverses any changes made to the current object and returns to the main menu

In a list of values [Cancel] stops the [List] function without selecting a value.

In the login screen [Cancel] exits SQL*Menu (Design).

[Change Display Type]

[Change Display Type] switches between 'forms and spread tables as the display type for object definitions, if both types are available.

In a form [Change Display Type] displays the spread table that corresponds to the current form and moves the cursor to the corresponding element. In the Application Definition form and the Option Selection form, [Change Display Type] has no function because there is no corresponding spread table.

In a spread table [Change Display Type] displays the form that corresponds to the current spread table and moves the cursor to the corresponding element.

[Clear Field]

[Clear Field] clears (deletes) the contents of the current field or line.

In a field [Clear Field] clears the entire field.

In a scroll region [Clear Field] clears the current line and rewraps text if possible.

[Copy]

[Copy] stores a copy of the selected text in the paste buffer until you use [Cut] or [Copy] again. [Select] marks one endpoint of the text and [Copy] marks the other endpoint. [Paste] inserts the contents of the paste buffer into any scroll region or field, truncating if necessary.

[Copy Object]

[Copy Object] displays the Copy/Reference Object dialog box, which allows you to copy or reference objects from any application to the current application. Referenced objects cannot be modified, but reflect any modifications made in their originating application.

[Cut]

[Cut] removes the selected text and stores it in the paste buffer until you use [Copy] or [Cut] again. [Select] marks one endpoint of the text and [Cut] marks the other endpoint. [Paste] inserts the contents of the paste buffer into any scroll region or field, truncating if necessary.

[Delete Backwards]

[Delete Backwards]:

- functions in fields and scroll regions
- deletes the character to the left of the current cursor position and moves any characters on the right one space to the left

[Delete Character]

[Delete Character]:

- functions in fields and scroll regions
- deletes the character at the current cursor position and moves any characters on the right one space to the left

[Delete Line]

[Delete Line]:

- functions only in scroll regions
- deletes the current line, and rewraps text if possible

[Delete Record]

In a form or spread table [Delete Record] deletes the current record from the form or spread table. The record is deleted from the database the next time you save the application.

[Down]

[Down] moves the cursor down one line.

In a menu [Down] moves the cursor to the next item in a pull-down menu. After the cursor reaches the last item, it returns to the first one.

[End of Line]

[End of Line] moves the cursor to the end of the current line.

In a field [End of Line] moves the cursor to the right of the last character in the current field, scrolling if necessary.

In a scroll region [End of Line] moves the cursor to the right of the last character in the current line.

[First Line]

[First Line] moves the cursor to the beginning of the first line (displayed or undisplayed) of the current region.

[Help]

In the design interface [Help] activates the online help system, displaying help for the current interface element or error message.

In the online help system [Help] displays the overview entitled "About the Online Help System."

[Insert Record]

[Insert Record]:

- functions in forms and spread tables
- creates a new (blank) record after the current record

[Insert/Replace]

[Insert/Replace] switches typing mode between Insert and Replace:

- In Replace mode, newly typed characters replace any existing characters, beginning at the current cursor position.
- In Insert mode, newly typed characters are inserted at the current cursor position, pushing the character already at that position (and any characters after it) to the right.

[Insert/Replace] functions in any data entry context. The status line shows the current data-entry mode with `<Ins>` or `<Rep>`.

[Last Line]

[Last Line] moves the cursor to the end of the last line (displayed or undisplayed) of the current region.

[Left]

In a main menu [Left]:

- highlights the menu item to the left of the current choice
- moves the highlighting to the rightmost menu item if the current item is the leftmost item

In a form or spread table [Left] moves the cursor one character to the left.

[List]

[List] displays a list of the possible values for the current field, if such a list of values exists. If the list is already displayd, [List] moves the cursor to the list.

The symbol `<List>` on the status line indicates that a list of values is available for the current field.

For more information about lists of values, see “List of Values” in Chapter 6.

[Menu]

[Menu] returns to the main menu of the design interface or moves the cursor to the menu of the online help system.

[Navigate]

[Navigate] displays the Navigate dialog box, which allows you to move to the spread table or form for a specific object (menu, item, parameter, or procedure).

Note: Use [Navigate] as a shortcut for directly accessing the form or spread table for a particular object. This feature is especially useful for designers working on block mode terminals because it usually requires fewer screen interrupts than alternative methods.

[Next Field]

In a menu [Next Field]:

- moves the cursor down or to the right, to the next item in a menu
- moves the cursor back to the first menu item from the last one

In a form or spread table [Next Field]:

- moves the cursor in sequence, to the next field, check box, button, or scroll region
- moves the cursor from the last element of a form or spread table back to the first element

In a list of values [Next Field] moves the cursor down, to the next value in the list of values.

[Next Record]

In a form [Next Record]:

- displays the data for the next record in all fields, check boxes, and scroll regions in the form
- displays a new record if the current record is the last record
- does nothing if the current record is a new record

In a spread table [Next Record]:

- moves the cursor down one line in a spread table, to the next record or to a new record
- does nothing if the cursor is in a new (blank) record

[Paste]

[Paste] inserts the contents of the paste buffer at the cursor's position. The paste buffer contains the text that you cut or copied most recently, using [Cut] or [Copy]. [Paste]:

- functions only in fields and scroll regions
- inserts a deleted or copied portion of text in the screen
- positions the upper left comer of the pasted area at the cursor position

[Previous Field]

In a menu [Previous Field]:

- moves the cursor up or to the left to the previous item in a menu
- moves the cursor to the last menu item if the cursor is at the first one

In a form or spread table [Previous Field]:

- moves the cursor in sequence to the previous field, check box, button, or scroll region in a form or spread table
- moves the cursor from the first element of a form or spread table to the last element

In a list of values [Previous Field] moves the cursor up to the previous value in the list of values.

[Previous Record]

In a form [Previous Record]:

- displays the data for the previous record in all fields, check boxes, and scroll regions in the form
- does nothing if the cursor is in the first record

In a spread table [Previous Record]:

- moves the cursor up one line in a spread table to the previous record
- does nothing if the cursor is in the first record

[Print]

[Print] displays the Print dialog box, which allows you to create an output file for the current screen.

[Quit]

[Quit] exits SQL*Menu (Design) and returns to the operating system.

[Refresh]

[Refresh] restores the screen image after it has been changed by some external cause (for example, by an operator message). [Refresh]:

- refreshes the current screen
- redisplays any changes you have made to the screen if you are on a character mode terminal
- discards any changes you have made to the screen and redisplays previous values if you are on a block mode terminal. (This is a terminal function, not a SQL*Menu function, on block mode devices.)

[Right]

In a main menu [Right]:

- highlights the menu item to the right of the current selection
- moves the highlight from the rightmost menu item to the leftmost item

In a form or spread table [Right] moves the cursor one character position to the right.

[Scroll Down]

In a scroll region [Scroll Down]:

- scrolls the contents of the scroll region up by approximately 80 percent of the scroll region's display, displaying lines of information that were "below" the scroll region's display
- navigates to the same column in the topmost displayed line once scrolling is complete

In a spread table [Scroll Down]:

- scrolls the contents of the spread table up by approximately 80 percent of the spread table's display, displaying records that were "below" the spread table's display
- navigates to the topmost displayed record once scrolling is complete

In a list of values [Scroll Down]:

- scrolls the contents of the list up by approximately 80 percent of the list's display, displaying values that were "below" the list's display
- navigates to the topmost displayed value once scrolling is , complete

[Scroll Left]

In a field [Scroll Left] scrolls approximately 80 percent of the displayed contents of a field to the right, effectively displaying data that was "to the left" of the field's display window.

[Scroll Right]

In a field [Scroll Right] scrolls approximately 80 percent of the displayed contents of a field to the left, effectively displaying data that was "to the right" of the field's display window.

[Scroll Up]

In a scroll region [Scroll Up]:

- scrolls the contents of the scroll region down by approximately 80 percent of the scroll region's display, displaying lines of information that were "above" the scroll region's display
- navigates to the same column in the topmost displayed line once scrolling is complete

In a spread table [Scroll Up]:

- scrolls the contents of the spread table down by approximately 80 percent of the spread table's display, displaying records that were "above" the spread table's display
- navigates to the top-most displayed record once scrolling is complete

In a list of values [Scroll Up]:

- scrolls the contents of the list down by approximately 80 percent of the list's display, displaying values that were "above" the list's display
- navigates to the topmost displayed value once scrolling is complete

[Search]

In a scroll region [Search] displays the Search dialog box, which allows you to search for, and optionally replace, a text string.

[Select]

In a menu [Select] chooses the indicated menu item.

In a check box [Select] toggles the value in a check box, thereby activating the previously inactive value.

In a set of radio buttons [Select] chooses the next radio button in the set, or returns from the last one to the first.

In a keyword in the Online Help System [Select] chooses the keyword the cursor is on in the help screen and navigates to the help screen for that keyword.

In a list of values [Select] chooses the value the cursor is on and closes the list of values.

[Show Keys]

[Show Keys] displays the keyboard map that corresponds to the terminal definition you specified (or to which the system defaulted) when you logged in to SQL*Menu.

Note: [Show Keys] only displays mappings for the keys that are available in the current context.

[Up]

[Up] moves the cursor up one line.

In a menu [Up] moves the cursor to the previous item in a pull-down menu. After the cursor reaches the first item, it skips to the last one.

[Zoom In]

[Zoom In] displays the form or spread table for the object type that is one step lower in the object hierarchy than the current object type. When you use [Zoom In], SQL*Menu moves down the following hierarchy of objects, starting from the level of the current object:

- application
- menu
- item

As SQL*Menu moves down the hierarchy, it maintains the current spread table or form display style where possible.

[Zoom Out]

[Zoom Out] displays the form or spread table for the object type that is one step higher in the object hierarchy than the current object type. When you use [Zoom Out], SQL*Menu moves up the following hierarchy of objects, starting from the level of the current object:

- application
- menu
- item

As SQL*Menu moves up the hierarchy, it maintains the current spread table or form display style where possible.

Functional Groupings for Designers

This section places the SQL*Menu (Design) function keys into groups by their basic characteristics:

- cursor movement functions
- editing functions
- general functions
- user assistance functions

Use these groupings to help locate a function when you know what you want to do, such as moving the cursor or deleting data. Once you find the name of the function key for which you want information, you can find its description in the alphabetical listing and find further information by using this book's index.

Note: Some keys appear in more than one group.

Cursor Movement Functions

Beginning of Line]	[Next Field]	[Scroll Down]
[Down]	[Next Record]	[Scroll Left]
[End of Line]	[Previous Field]	[Scroll Right]
[First Line]	[Previous Record]	[Scroll Up]
[Last Line]	[Right]	[Up]
[Left]		

Editing Functions

[Beginning of Line]	[Delete Backwards]	[Insert/Replace]
[Clear Field]	[Delete Character]	[Left]
[Copy]	[Delete Line]	[Paste]
[Cut]	[End of Line]	[Right]
[Delete Record]	[Insert Record]	[Search]

General Functions

[Accept]	[Quit]
[Cancel]	[Select]

User Asistance Functions

[Bookmark]	[List]	[Refresh]
[Change Display Type]	[Menu]	[Show Keys]
[Copy Object]	[Navigate]	[Zoom In]
[Help]	[Print]	[Zoom Out]

IV SQL*MENU REFERENCE

CHAPTER

8

SQL*MENU OBJECTS

This chapter describes the objects that make up a SQL*Menu application. It covers the following topics about SQL*Menu objects:

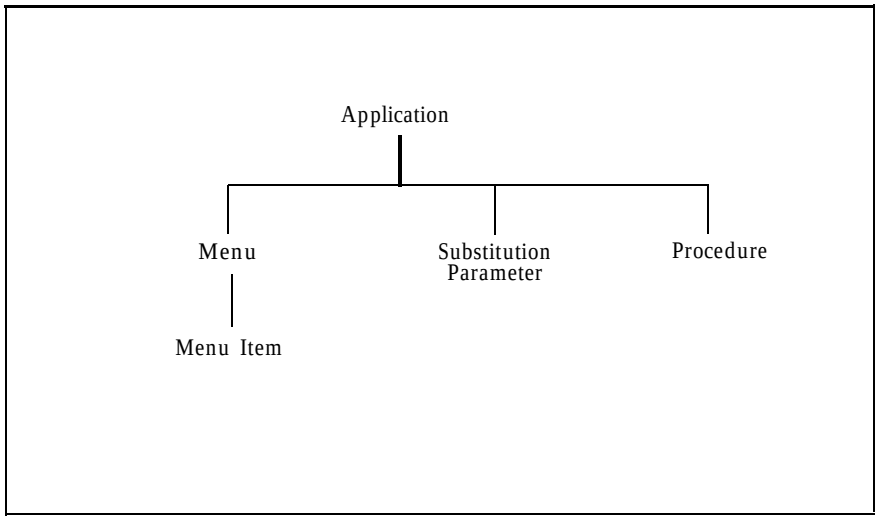
- object hierarchy
- object-naming conventions
- applications
- menus
- menu items
- substitution parameters
- procedures

Object Hierarchy

An *object* is a named group of data in the ORACLE database that you can copy, move, or delete as one entity in a single operation. Each type of object has default rules and values that you can override or extend for individual objects.

An object can own other objects, or belong to another object, in a *hierarchy of objects.* Figure 8-1 illustrates this hierarchy and shows how higher-level objects own lower-level objects.

FIGURE 8-1
The SQL*Menu Object Hierarchy



The current context within the object hierarchy determines which instances of each type of object can be accessed at a given time. For example, in the design interface an application must be established in the current context before a menu can be defined, because every menu belongs to an application.

Substitution parameters and procedures belong directly to an application. Although they are used by menu items, they are defined independently of the menu and item context.

Object-Naming Conventions

SQL*Menu object names, like ORACLE object names, must follow these conventions:

- can be up to 30 characters long
- must begin with a letter
- can contain letters, numbers, and special characters $,#, @,_
- cannot duplicate the name of another object of the same type in the same context
- cannot duplicate a reserved word

SQL*Menu does not distinguish between uppercase and lowercase letters in object names. For example, if you name one menu “Last” then you cannot name another menu "LaST" in the same application.

File Names

Application libraries are stored in an operating system file as the name of the menu application plus an extension *(application_name.* DMM). Therefore, in addition to following the standard object-naming conventions, the application filenames must follow the file-naming rules of your operating system.

Reserved Words

Do not use any of the ORACLE, PL/SQL, or SQL*Forms reserved words to name an object. Table 10-1 lists the reserved words.

TABLE 10-1
Reserved Words

ABORT	AUDIT	CLUSTERS	CURSOR
ACCEPT	AUTHORIZATION	COLAUTH	DATABASE
ACCESS	AVG	COLUMN	DATA_BASE
ADD	BEGIN	COLUMNS	DATAPAGES
ALL	BETWEEN	COMMENT	DATE
ALTER	BLOCK	COMMIT	DBA
AND	BODY	COMPRESS	DEBUGOFF
ANY	BOOLEAN	CONNECT	DEBUGON
APPEND	BY	CONSTANT	DECIMAL
ARRAY	CASE	CONTAIN	DECLARE
AS	CHAR	CONTAINS	DEFAULT
ASC	CHAR_BASE	COUNT	DEFINITION
ASSERT	CHECK	CRASH	DELAY
ASSIGN	CLOSE	CREATE	DELETE
AT	CLUSTER	CURRENT	DELTA

Table continued on next page... .

TABLE 10-1
Reserved Words
(Continued)

DESC	INDEXES	ORDER	SQLERRM
DIGITS	INDEXPAGES	OTHERS	START
DISPOSE	INDICATOR	OUT	STATEMENT
DISTINCT	INITIAL	PACKAGE	STDDEV
DO	INSERT	PARTITION	SUBTYPE
DOES	INTEGER	PCTEREE	SUCCESSFUL
DROP	INTERSECT	PRAGMA	SUM
EACH	INTO	PRIOR	SYNONYM
ELSE	IS	PRIVATE	SYSDATE
ELSIF	LEVEL	PRIVILEGES	SYSSORT
END	LIKE	PROCEDURE	SYSTEM
ENTRY	LIMITED	PUBLIC	TABAUTH
ERASE	LIST	RAISE	TABLE
EVALUATE	LOCK	RANGE	TABLES
EXCEPTION	LONG	RAW	TASK
EXCEPTION_INIT	LOOP	RECORD	TEMPORARY
EXCLUSIVE	MAX	RELEASE	TERMINATE
EXISTS	MAXEXTENTS	REM	THEN
EXIT	MIN	RENAME	TO
FALSE	MINUS	RENAMES	TRIGGER
FETCH	MOD	RESOURCE	TRUE
FIELD	MODE	RETURN	TYPE
FILE	MODIFY	REVERSE	UID
FLOAT	MOVE	REVOKE	UNION
FOR	NEW	ROLLBACK	UNIQUE
FORM	NOAUDIT	ROW	UPDATE
FORMAT	NOCOMPRESS	ROWID	USE
FROM	NOLIST	ROWNUM	USER
FUNCTION	NOSYSSORT	ROWS	USING
GENERIC	NOT	ROWTYPE	VALIDATE
GLOBAL	NOWAIT	RUN	VALUES
GOTO	NULL	SAVEPOINT	VARCHAR
GRANT	NUMBER	SCHEMA	VARGRAPHIC
GRAPHIC	NUMBER.BASE	SELECT	VARIANCE
GROUP	OF	SEPARATE	VIEW
HAVING	OFFLINE	SESSION	VIEWS
IDENTIFIED	OLD	SET	WHEN
IF	ON	SHARE	WHENEVER
IMAGE	ONLINE	SIZE	WHERE
IMMEDIATE	OPEN	SMALLINT	WHILE
IN	OPTIMIZE	SPACE	WITH
INCREMENT	OPTION	SQL	WORK
INDEX	OR	SQLCODE	XOR
INDEXED			

Applications

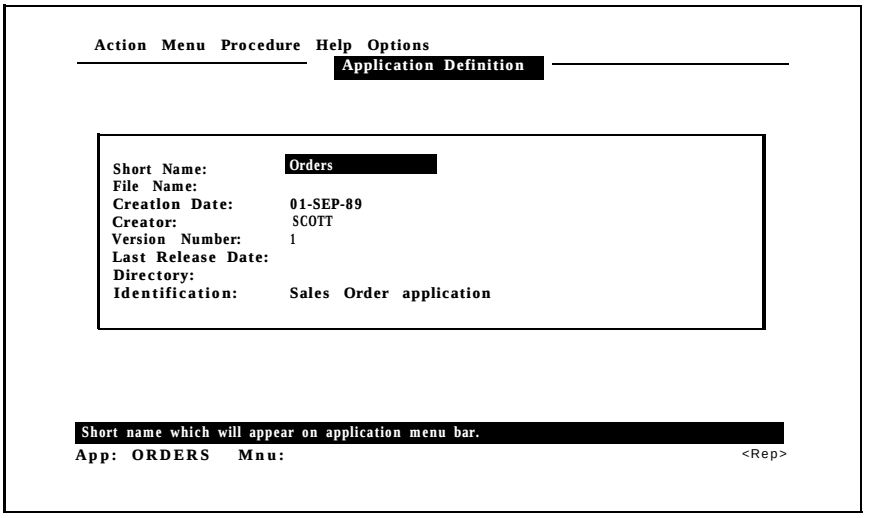
This section describes the *application* object in SQL*Menu. The application object is at the top of the object hierarchy. Application objects contain menu, parameter, and procedure objects.

Defining Applications

You create new applications in the SQL*Menu (Design) interface through the New Menu Application dialog box, and you modify application definitions through the Application Definition form, as described in "Basic Design Operations" in Chapter 5.

The Application Definition form shows information for the current application (Figure 8-2). To change the current application, use the Open Menu Application dialog box as described in "Opening a Menu Application" in Chapter 5.

FIGURE 8-2
Application Definition Form



Location: To reach the Application Definition form from the main menu, complete the following steps

1. Make sure you have application context. (You can also have menu and item context, but it is not necessary.)
2. Select the Menu item from the main menu of the design interface.
3. In the Menu submenu, select the Application item.

You can also display the Application Definition form by pressing

- [Zoom Out] from the Menu Definition form or spread table
- [Navigate] from any form or spread table, or from any menu of the design interface, and then pressing [Accept] without specifying any context

Application Characteristics

The process of defining an application consists of setting each of the application's characteristics. These characteristics indicate the following things about an application:

- identifying information
- information about the creation of the application
- location of the application library file
- version number and release date

This section presents application characteristics in alphabetical order. The names of application characteristics correspond exactly with the names of the elements in the Application Definition form, except for the Application Name characteristic, which is not an Application Definition element. For example, the Creator field in the Application Definition form defines the Creator application characteristic.

You assign application characteristics by typing values into field elements.

Application Name

The Application Name application characteristic specifies the name of the SQL*Menu application. You assign the Application Name when you create a new application or open an existing application.

Required/Optional: required

Creation Date

The Creation Date application characteristic specifies the date that the application is created, in the form DD-MON-YY.

The default value is the current date.

Required/Optional: required

Creator

The Creator application characteristic specifies the username of the application creator or modifier.

The default value is the current ORACLE username.

Required/Optional: required

Directory

The Directory application characteristic specifies the directory path for the application library file. SQL*Menu (Generate) creates the library file in this directory. If you do not specify a directory, SQL*Menu uses the current directory.

SQL*Menu (Run Menu) searches the specified directory for the library file when it runs the application. If you do not specify a directory, SQL*Menu searches the current directory. If SQL*Menu cannot find the file in the current directory, it searches a predefine path. You can define this path for every platform on which you run the application. Refer to the appropriate *ORACLE Installation and User's Guide* for more information about this path.

Required/Optional: optional

File Name

The File Name application characteristic specifies a name for the file containing the application library. The name does not include the file extension, which SQL*Menu adds when it creates *filename.* DMM.

The default value is the Application Name.

Required/Optional: required

Identification

The Identification application characteristic specifies a description of the application. This 40-character string identifies the application in a full-screen display of the application menu. In pull-down and bar displays of the application menu, the Identification appears in the message line when the cursor is on the application.

The default value is the Application Name.

Required/Optional: required

Last Release Date

The Last Release Date application characteristic specifies the date of the application's last release, in the form DD-MON-YY.

Required/Optional: optional

Short Name

The Short Name application characteristic specifies a short name for use in the application menu of bar and pull-down menu display styles.

The default value is the Application Name, truncated to 15 characters if necessary.

Required/Optional: required

Version Number

The Version Number application characteristic specifies the version and release number. You can use whatever numbering system you want to keep track of different versions of an application.

Required/Optional: required

Menus

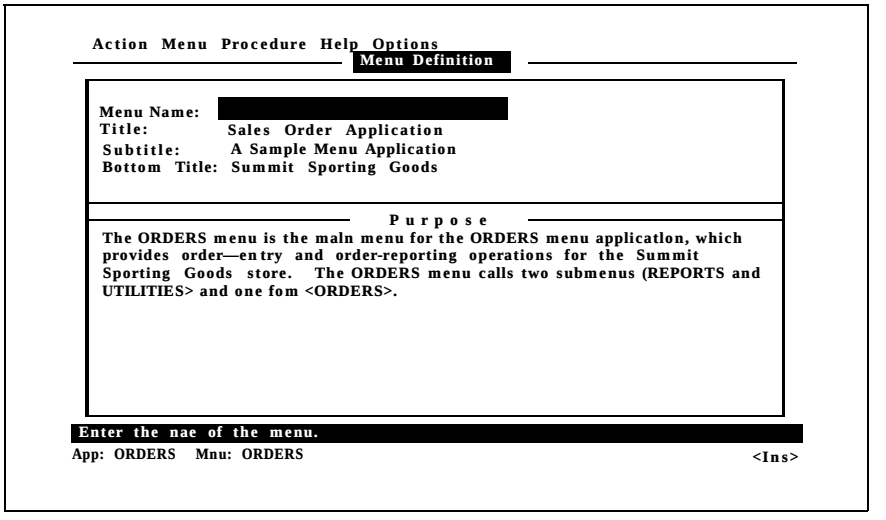
This section describes the *menu* object in SQL*Menu. The menu object is below the application object in the object hierarchy, and above the menu item object.

Defining Menus

You create and modify menus in the SQL*Menu (Design) interface through the Menu Definition form or spread table, as described in "Defining a Menu" in Chapter 5.

The Menu Definition form shows information for one menu at a time (Figure 8-3), and the Menu Definition spread table shows information for several menus at once (Figure 8-4).

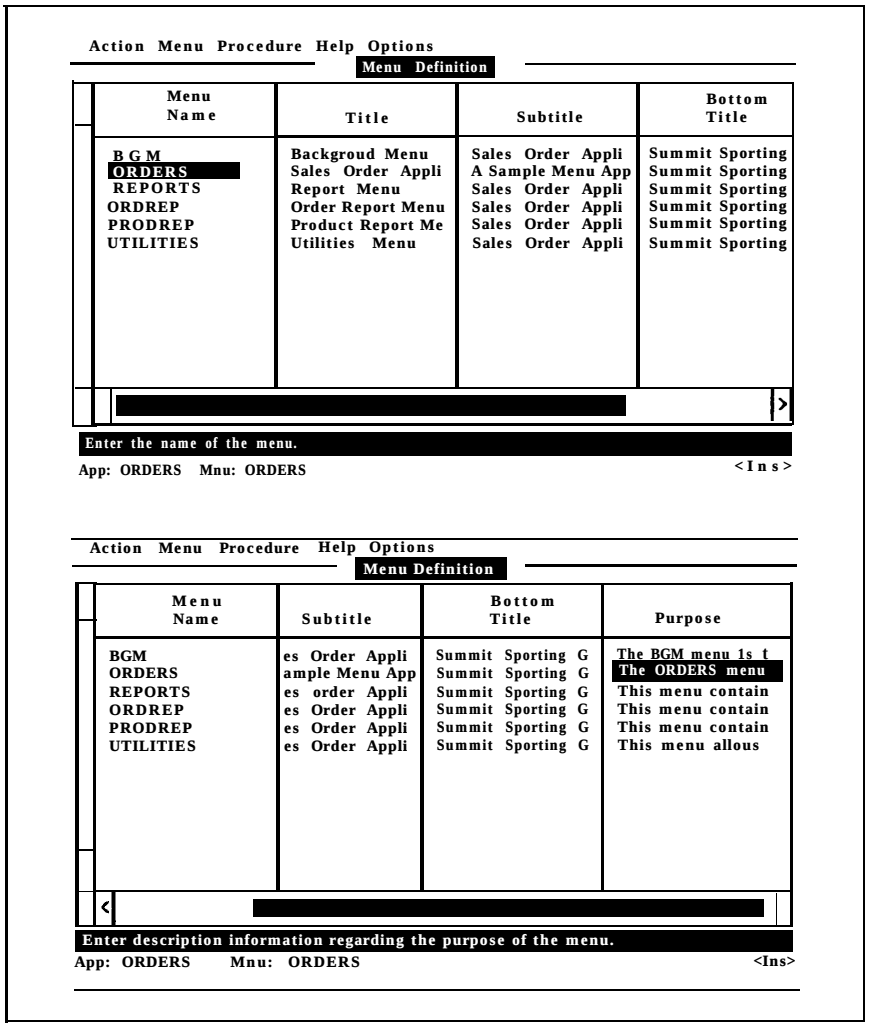
FIGURE 8-3
Menu Definition Form



Location: To reach the Menu Definition form or spread table from the main menu, complete the following steps:

1. Make sure you have application context. (You can also have menu and item context, but it is not necessary.)
2. Select the Menu item from the main menu of the design interface.
3. In the Menu submenu, select the Menu item.
4. Press [Change Display Type] to toggle the form and spread table display types.

FIGURE 8-4
Menu Definition Spread Table



You can also display the Menu Definition form or spread table by pressing

- [Zoom In] from the Application Definition form
- [Zoom Out] from the Item Definition form or spread table
- [Navigate] from any form or spread table, or from any menu of the design interface, then specifying a menu context

Menu Characteristics

The process of defining a menu consists of setting each of the menu's characteristics. These characteristics indicate the following things about a menu:

- identifying information
- display information

This section presents menu characteristics in alphabetical order. The names of menu characteristics correspond exactly with the names of the elements in the Menu Definition form and the Menu Definition spread table.

You assign menu characteristics by typing values into field and scroll region elements.

Bottom Title

The Bottom Title menu characteristic specifies a title of up to 72 characters to appear at the bottom of the menu, above the message line and status line, in full-screen menus.

The Bottom Title is not displayed in pull-down or bar menus.

Required/Optional: optional

Menu Name

The Menu Name menu characteristic specifies a name of up to 30 characters for the menu. The main menu must have the same name as the application.

The Menu Name appears in the status line and identifies the menu for navigation in full-screen menus. The Where option displays all the menu names in the current branch of the menu tree.

Required/Optional: required

Purpose

The Purpose menu characteristic specifies the purpose of the menu. You assign the Purpose menu characteristic in a scroll region in the Menu Definition form or spread table.

The information in the Purpose scroll region does not appear on screen. Use this scroll region for comments or other documentation about the menu and its contents.

Required/Optional: required

Subtitle

The Subtitle menu characteristic specifies a subtitle of up to 40 characters to appear below the Title in full-screen menus.

The Subtitle is not displayed in pull-down or bar menus.

Required/Optional: required

Title

The Title menu characteristic specifies a title of up to 40 characters to appear in full-screen menus at the top of the menu.

The Title is not displayed in pull-down or bar menus.

Required/Optional: required

Menu Items

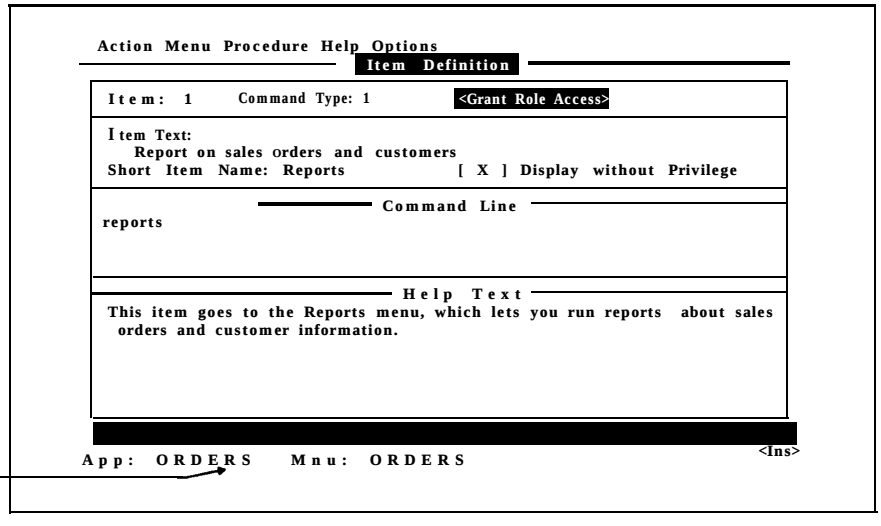
This section describes the *menu item* object in SQL*Menu. The menu item object is below the menu object in the object hierarchy.

Defining Menu Items

You create and modify menu items in the SQL*Menu (Design) interface through the Item Definition form or spread table, as described in "Defining a Menu Item" in Chapter 5.

The Item Definition form shows information for one menu item at a time (Figure 8-5), and the Item Definition spread table shows information for several menu items at once (Figure 8-6).

FIGURE 8-5
Item Definition Form

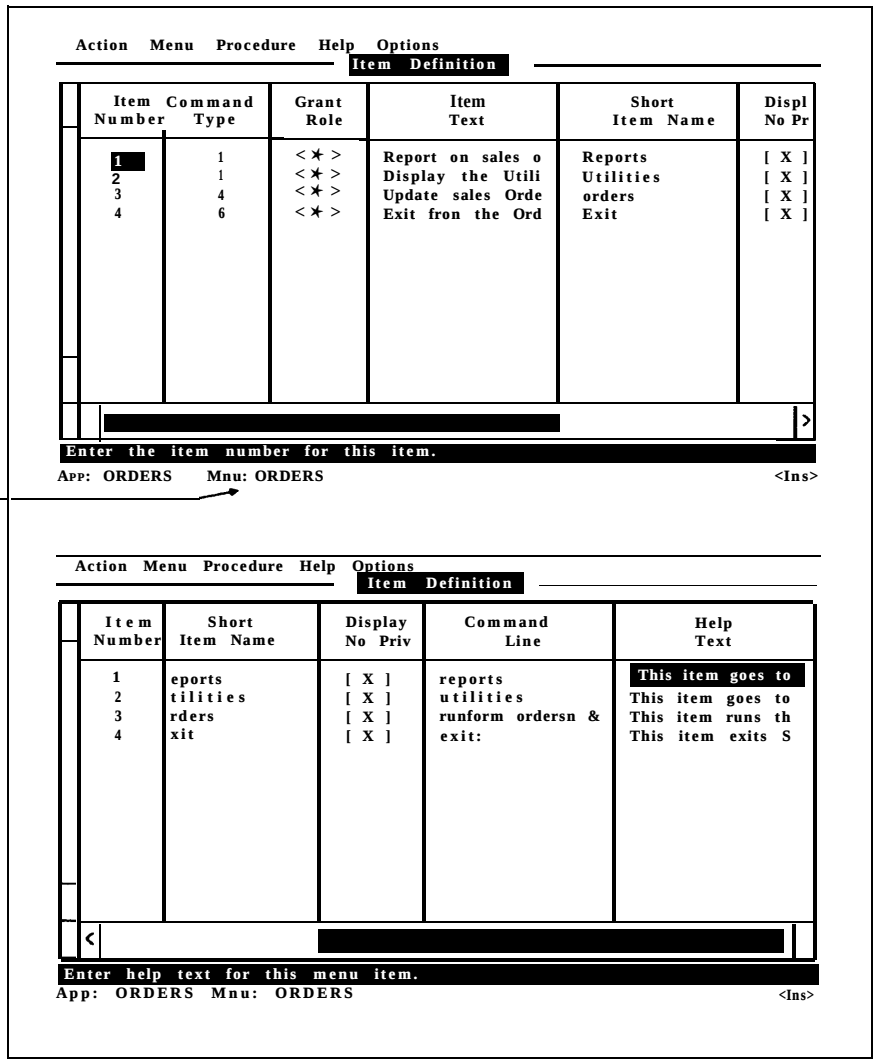


Location: To reach the Item Definition form or spread table from the main menu, complete the following steps:

1. Make sure you have application and menu context. (You can also have menu item context, but it is not necessary.)
2. Select the Menu item from the main menu of the design interface.
3. Select Item in the Menu submenu.

4. Press [Change Display Type] to toggle the form and spread table display types.

FIGURE 8-6
Item Definition Spread Table



Status line shows the current menu name

You can also display the Item Definition form or spread table by pressing:

- [Zoom In] from the Menu Definition form or spread table
- [Navigate] from any form or spread table, or from any menu of the design interface, and then specifying a menu item context

Menu Item Characteristics

The process of defining a menu item consists of setting each of the menu item's characteristics. These characteristics indicate the following things about a menu item:

- identifying information
- display information
- command information

This section presents menu item characteristics in alphabetical order. The names of menu item characteristics correspond exactly with the names of the elements in the Item Definition form, and correspond approximately with the names of elements in the Item Definition spread table. For example, the Display without Privilege check box in the Item Definition form corresponds to the Display No Priv check box in the Item Definition spread table.

You assign menu item characteristics by typing values into field, check box, and scroll region elements, or by selecting a button to display the Item Role pop-up window.

Command Line

The Command Line menu item characteristic specifies a menu name, one or more packaged procedures or macros, or a command statement, as described in the following section, "SQL*Menu Command Types." The Command Line value can be up to 240 characters long.

You assign the Command Line menu item characteristic in a scroll region in the Item Definition form or spread table.

Required/Optional: required

Command Type

The Command Type menu item characteristic specifies a number from 1 to 7 for the type of command statement in the Command Line, as described in the following section, "SQL*Menu Command Types."

Required/Optional: required

Display without Privilege

The Display without Privilege menu item characteristic specifies whether to display the menu item to users who do not have access to use it. (Inaccessible items appear without highlighting or bold characters.) You assign the Display without Privilege menu item characteristic in a check box in the Item Definition form or spread table.

The default setting is off, only displaying the menu item to authorized users.

Required/Optional: required

Grant Role Access

The Grant Role Access menu item characteristic specifies the names of existing roles that need access to this menu item. You assign the Grant Role Access menu item characteristic in a pop-up window (the Item Role window) that appears when you select the Grant Role Access button in the Item Definition form or spread table.

If you do not explicitly assign any roles to a menu item, all roles that have access to the application have access to the item. Once you explicitly assign one role to an item (by entering it in the Item Role window), only roles that you specify in this characteristic will have access to the item.

Required/Optional: optional

Help Text

The Help Text menu item characteristic specifies the help text for the menu item. You assign the Help Text menu item characteristic in a scroll region in the Item Definition form or spread table.

This text will appear in a window if the operator presses [Help] while the menu item is selected.

Required/Optional: optional

Item

The Item menu item characteristic specifies a number that determines the order of menu items within the menu. In the full-screen menu display style, this number is used for navigation.

The default value is one greater than the number of the menu item in the previous record.

Required/Optional: required

Item Text

The Item Text menu item characteristic specifies a description of up to 70 characters. This text appears in the message line for pull-down and bar menus, or next to the item number in full-screen menus.

Required/Optional: required

Short Item Name

The Short Item Name menu item characteristic specifies a description of up to 15 characters to appear as the menu item in pull-down and bar style displays.

Required/Optional: required

SQL*Menu Command Types

This section describes the seven command types available in SQL*Menu Version 5.0. To use these commands, enter the command type number into the Command Type field of the Item Definition form or spread table and then enter an appropriate command in the Command Line scroll region.

Table 8-1 lists the SQL*Menu command types.

TABLE 8-1
Command Types

Type	*Description*
1	Invoke a submenu
2	Execute an operating system command
3	Execute an operating system command, then pause for the operator's response
4	Invoke SQL*Forms (Run Form)
5	Invoke SQL*Plus
6	Execute a SQL*Menu macro command
7	Execute a PL/SQL command

Type 1: Invoking a Submenu

Command type 1 lets you attach a submenu to a menu item. When an operator selects that item, SQL*Menu displays the menu specified in the item's Command Line.

In the pull-down and bar menu display styles, the submenu cannot be above the current menu in the same branch of the menu tree. In general, you should make your menu tree a simple hierarchy.

Table 8-2 shows the items in the sample Orders Application that use command type 1 to invoke a submenu.

TABLE 8-2
Type 1 Commands in the Orders Application

Menu Name	*Item Number*	*Command Type*	*Command Line (Menu Invoked)*
ORDERS	1	1	REPORTS
ORDERS	2	1	UTILITIES
REPORTS	1	1	ORDREP
REPORTS	2	1	PRODREP

Types 2 and 3: Executing Operating System Commands

Command types 2 and 3 pass the value of the Command Line to the operating system for processing. After executing the command, type 2 returns to SQL*Menu immediately. Type 3 pauses after executing the command and waits for the operator to press a key before returning to SQL*Menu.

These command types can be used to invoke other software products, such as SQL*ReportWriter, SQL*Forms, and SQL*Plus. If SQL*Forms or SQL*Plus is linked with SQL*Menu, however, command type 4 or 5 will invoke the linked program more quickly.

Table 8-3 shows the item in the sample Orders Application that uses command type 3 to execute an operating system command that lists the contents of the default directory. The substitution parameter &ST specifies a search string, with* as the default value.

TABLE 8-3
Type 3 Command in the Orders Application

Menu Name	*Item Number*	*Command Type*	*Command Line (OS Command)*
UTILITIES	1	3	`DIR &ST`

This example uses command type 3 rather than type 2 so that the operator can review the directory listing during the pause before returning to SQL*Menu.

Type 4: Invoking SQL*Forms

SQL*Forms and SQL*Menu may be linked together when SQL*Menu is installed. If this option is selected, SQL*Menu uses menu command type 4 to invoke a form without a new database login. This invocation of a form is much quicker than invocation by command type 2 or 3.

Command type 4 executes as command type 2 if SQL*Menu has not been linked with SQL*Forms at installation.

The Command Line scroll region contains any information for the SQL*Forms login, just as if you were invoking SQL*Forms from the operating system level. Values such as username, password, command file, and arguments can be passed by using substitution parameters, for which the user must supply values.

To specify a form name and SQL*Forms options, simply enter the command line as if you were invoking SQL*Forms from the operating system prompt. For example, to run a form called EMP, enter `RUNFORM EMP &UN/ &PW` into the Command Line scroll region.

Note: If SQL*Forms and SQL*Menu are linked, SQL*Forms uses the current login ID regardless of whether you specify a username and password in the command line. Specifying `&UN/&PW` is a precaution in case SQL*Forms and SQL*Menu are not linked and command type 4 executes as type 2. Also, the SQL*Forms (Run Form) command line switches -c, -e, -i, -r, and -w are not supported in command type 4 when SQL*Forms and SQL*Menu are linked. To specify one of these switches, you can use command type 2 or 3.

Table 8-4 shows the items in the sample Orders Application that use command type 4 to run the ORDERS sample form. The directory represented by *path* varies for different operating systems. (The background menu item duplicates the item in the main menu.)

TABLE 8-4
Type 4 Commands in the Orders Application

Menu Name	*Item Number*	*Command Type*	*Command Line (Form Invoked)*
ORDERS	3	4	`RUNFORM` *`pathORDERS`* `&UN/&PW`
BGM	2	4	`RUNFORM` *`pathORDERS`* `&UN/&PW`

Global Variables in SQL*Forms SQL*Menu preserves any global variables defined by SQL*Forms when using command type 4 (only if SQL*Forms is linked with SQL*Menu). Thus, forms can define global variables and then refer back to them.

Type 5: Invoking SQL*Plus

SQL*Plus and SQL*Menu may be linked together when SQL*Menu is installed. If this option is selected, SQL*Menu uses menu command type 5 to invoke SQL*Plus without a new database login and without loading SQL*Plus into memory. Command type 5 invokes SQL*Plus significant y faster than command type 2 or 3.

Command type 5 executes as command type 2 if SQL*Menu has not been linked with SQL*Plus at installation.

The Command Line scroll region contains any information for the SQL*Plus login, just as if you were invoking SQL*Plus from the operating system level. Values such as username, password, command file, and arguments can be passed by using substitution parameters, for which the user must supply values.

For example, to invoke SQL*Plus and run a SQL script titled JOBS with the current username (&UN) and password (&PW), you enter the following in the Command Line

```
SQLPLUS &UN/&PW @JOBS
```

Note: If SQL*Plus and SQL*Menu are linked, SQL*Plus uses the current login ID regardless of whether you specify another username and password in the command line.

Table 8-5 shows the items in the sample Orders Application that use command type 5 to run reports in SQL*Plus. The directory represented by *path* varies for different operating systems. Substitution parameters represent the report's starting and ending dates (&SD and &ED) or customer identification code (&CI). (The background menu item duplicates the Dated Orders item in the ORDREP menu.)

TABLE 8-5 Type 5 Commands in the Orders Application

Menu Name	*Item Number*	*Command Type*	*Command Line (SQL*Plus Report Invoked)*
REPORTS	3	5	SQLPLUS -S &UN/&PW *@pathCCREDIT*
ORDREP	1	5	SQLPLUS -S &UN/&PW *@pathALLORDS*
ORDREP	2	5	SQLPLUS -S &UN/&PW *@pathDATORDS* &SD &ED
PRODREP	1	5	SQLPLUS -S &UN/&PW *@pathALLPROD*
PRODREP	2	5	SQLPLUS -S &UN/&PW *@pathCUSPROD* &CI
BGM	1	5	SQLPLUS -S &UN/&PW *@pathDATORDS* &SD &ED

Type 6: Executing Macro Commands

To use SQL*Menu macro commands, enter 6 into the Command Type field of the Item Definition form or spread table and enter one or more macros into the Command Line scroll region.

The following macros are available in SQL*Menu:

APLMENU	DISP	NEWUSER	RIGHT
APLPARM	DOWN	NODISP	SHOWBGM
ASSIGN	EXIT	NXTFLD	SHOWKEYS
BGM *n*	HELP	OSCMD	SUSPEND
CHRMODE	LEFT	OSCMD1	TRMNATE
CLRFLD	MAINMENU	PRVFLD	UP
DEBUG	MENUPARM	PRVMENU	WHERE
DELCHR	NEWAPL	REDISP	

These macros are discussed in Chapter 11, "SQL*Menu Macros."

Table 8-6 shows the item in the sample Orders Application that uses command type 6 to execute the OSCMD macro.

TABLE 8-6
Type 6 Command in the Orders Application

Menu Name	*Item Number*	*Command Type*	*Command Line (Macro)*
UTILITIES	2	6	`OSCMD;`

Type 7 ExecutingPL/SQL Commands

To execute a PL/SQL command, enter 7 into the Command Type field of the Item Definition form or spread table and enter the command into the Command Line scroll region. The command text can be either a procedure name or the text for an unnamed procedure. PL/SQL packaged procedures are available for SQL*Menu and SQL*Forms.

Table 8-7 shows the item in the sample Orders Application that uses command type 7 to execute a PL/SQL packaged procedure that exits SQL*Menu.

TABLE 8-7
Type 7 Command in the Orders Application

Menu Name	*Item Number*	*Command Type*	*Command Line (PU/SQL Command)*
ORDERS	4	7	`EXIT_MENU;`

For more information about PL/SQL commands, see Chapter 9, "PL/SQL," and Chapter 10, PL/SQL Packaged Procedures."

Substitution Parameters

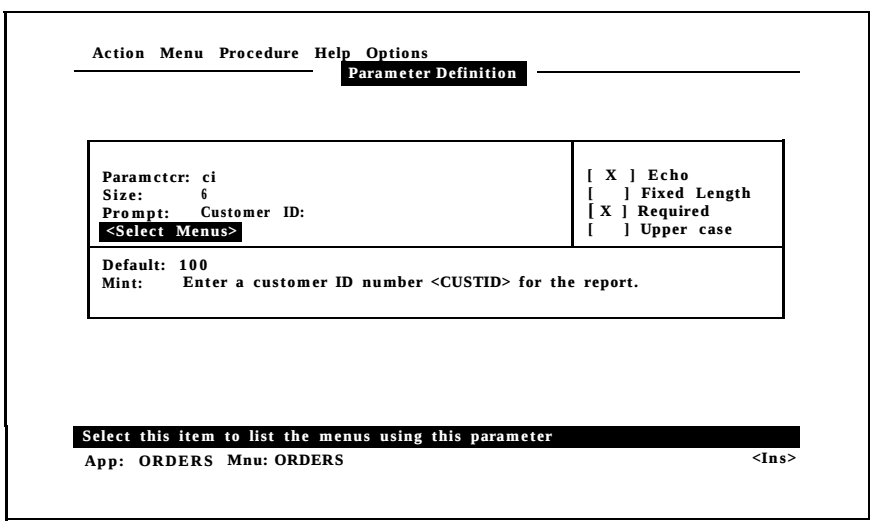
This section describes the *substitution parameter* object in SQL*Menu. A substitution parameter is identified by a two-character code that can stand for a value in the command line or in a procedure. The substitution parameter object is below the application object in the object hierarchy.

Defining Substitution Parameters

You create and modify substitution parameters in the SQL*Menu (Design) interface through the Parameter Definition form or spread table, as described in "Defining a Substitution Parameter" in Chapter 5.

The Parameter Definition form shows information for one parameter at a time (Figure 8-7), and the Parameter Definition spread table shows information for several parameters at once (Figure 8-8).

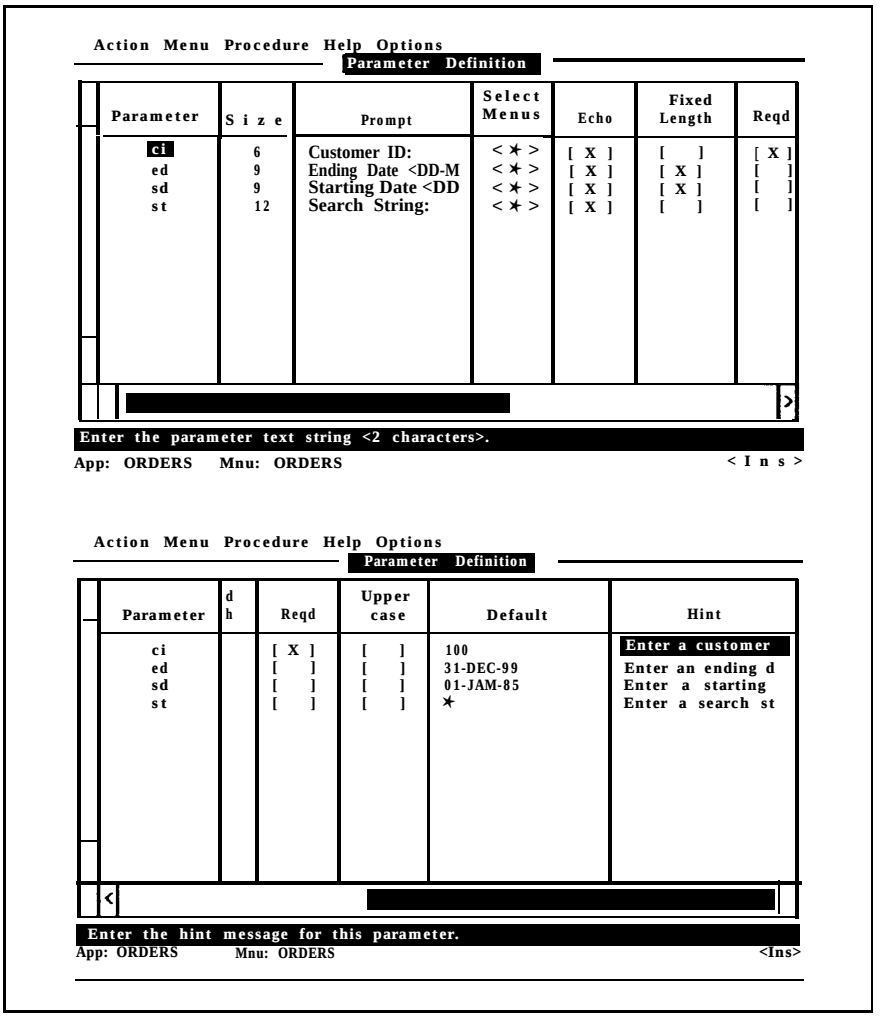
FIGURE 8-7
Parameter Definition Form



Location: To reach the Parameter Definition form or spread table from the main menu, complete the following steps:

1. Make sure you have application context. (You can also have menu and item context, but it is not necessary.)
2. Select the Menu item from the main menu of the design interface.
3. In the Menu submenu, select the Parameter item.
4. Press [Change Display Type] to toggle the form and spread table display types.

FIGURE 8-8
Parameter Definition Spread Table



You can also display the Parameter Definition form or spread table by pressing [Navigate] from any form or spread table, or from any menu of the design interface, then specifying a parameter context.

Parameter Characteristics

The process of defining a parameter consists of setting each of the parameter's characteristics. These characteristics indicate the following things about a parameter:

- identifying information
- display information
- default value
- associated menus

This section presents parameter characteristics in alphabetical order. The names of parameter characteristics correspond exactly with the names of the elements in the Parameter Definition form, and approximately with the names of elements in the Parameter Definition spread table. For example, the Required check box in the Parameter Definition form corresponds to the Reqd check box in the Parameter Definition spread table.

You assign parameter characteristics by typing values into field and check box elements, or by selecting a button to display the Menu Name pop-up window.

Default

The Default parameter characteristic specifies a default value for the substitution parameter.

Required/Optional: optional

Echo

The Echo parameter characteristic specifies whether to display the parameter value as the operator enters it. You assign the Echo parameter characteristic in a check box in the Parameter Definition form or spread table.

The default setting is on, which echoes characters as they are entered.

Required/Optional: required

Fixed Length

The Fixed Length parameter characteristic specifies whether the parameter value must have the maximum number of characters allowed by the Size characteristic. You assign the Fixed Length parameter characteristic in a check box in the Parameter Definition form or spread table.

The default setting is off, allowing any length up to the maximum.

Required/Optional: required

Hint

The Hint parameter characteristic specifies a description or instruction to appear in the message line when the operator is entering a value for the substitution parameter.

Required/Optional: optional

Parameter

The Parameter characteristic specifies a two-character name for the substitution parameter. The characters must be alphanumeric. In a command line reference the Parameter is preceded by an ampersand *(&xx)*, and in a PL/SQL reference it is preceded by a colon *(:xx)*.

Required/Optional: required

Prompt

The Prompt parameter characteristic specifies a label that will prompt the operator to supply a value for the substitution parameter.

Required/Optional: optional

Required

The Required parameter characteristic specifies whether a value must be supplied for the substitution parameter. You assign the Required parameter characteristic in a check box in the Parameter Definition form or spread table.

The default setting is off, allowing a NULL value.

Required/Optional: required

Select Menus

The Select Menus parameter characteristic specifies the names of menus that use this substitution parameter as a menu parameter in the full-screen display style. You assign the Select Menus parameter characteristic in a pop-up window (the Menu Name window) that appears when you select the Select Menus button in the Parameter Definition form or spread table.

Required/Optional: optional

Size

The Size parameter characteristic specifies the maximum number of characters that the operator can enter for the value of the substitution parameter.

Required/Optional: required

Upper Case

The Upper Case parameter characteristic specifies whether SQL*Menu converts the value for a substitution parameter to uppercase letters. You assign the Required parameter characteristic in a check box in the Parameter Definition form or spread table.

The default setting is off, which does not convert values.

Required/Optional: required

Using Substitution Parameters

SQL*Menu lets you construct flexible, general-purpose menus through the use of substitution parameters. For example, you can create a menu item that invokes SQL*Forms, using a substitution parameter that prompts for the name of the form the operator wants to run.

You can associate any substitution parameter with any menu in an application, making a *menu parameter*. When an operator selects that menu in the full-screen display style, the Enter Parameter Values dialog box prompts the operator to enter the exact value.

Note: Menu parameters are only useful for full-screen menus.

You can also associate a substitution parameter with a menu item, making an *item parameter*. When an operator selects that item, the Enter Parameter Values dialog box prompts for the exact value.

The Parameter Definition form or spread table can be used to create customized substitution parameters for menu applications. SQL*Menu also provides several standard substitution parameters, as described in the following section.

Standard Substitution Parameters

Five predefined substitution parameters, which are always known to SQL*Menu, are &UN, &PW, &AD, &SO, and &TT. They contain the following values:

&UN	the current username
&PW	the current password
&AD	the directory where the current application library file is stored
&SO	the current menu item (“selected option”)
&TT	the terminal type you used to log in to SQL*Menu

The ampersand (&) is required before the parameter name when you refer to a substitution parameter in the command line of a menu item. In a block of PL/SQL code, you must refer to parameters in the form `:UN`, `:PW`, and so on.

Parameter Processing

Before executing a command, SQL*Menu checks the command sting for substitution parameters. For each substitution parameter, SQL*Menu also determines whether it is a menu parameter for the current menu. (Menu parameters are only useful for full-screen menus.)

If the substitution parameter is not a menu parameter for the current menu, SQL*Menu checks whether it is a menu parameter for the main menu of the current application. Only when a substitution parameter is not for the current menu or main menu is it considered to be an item parameter. SQL*Menu then adds it to the item parameter list.

After processing each menu parameter in the command, SQL*Menu checks the item parameter list. If the list is not empty, SQL*Menu activates the Enter Parameter Values dialog box and the operator can enter a value. Otherwise, SQL*Menu executes the command after substituting the menu parameter values in the command string.

When SQL*Menu activates the Enter Parameter Values dialog box, the operator can change the item parameter value. After the Enter Parameter Values dialog box terminates, SQL*Menu executes the command using the new value.

Procedures

This section describes the *procedure* object in SQL*Menu. The procedure object is below the application object in the object hierarchy.

A procedure is a set of instructions that an application stores for repeated execution. Typically, a procedure consists of SQL statements and PL/SQL commands.

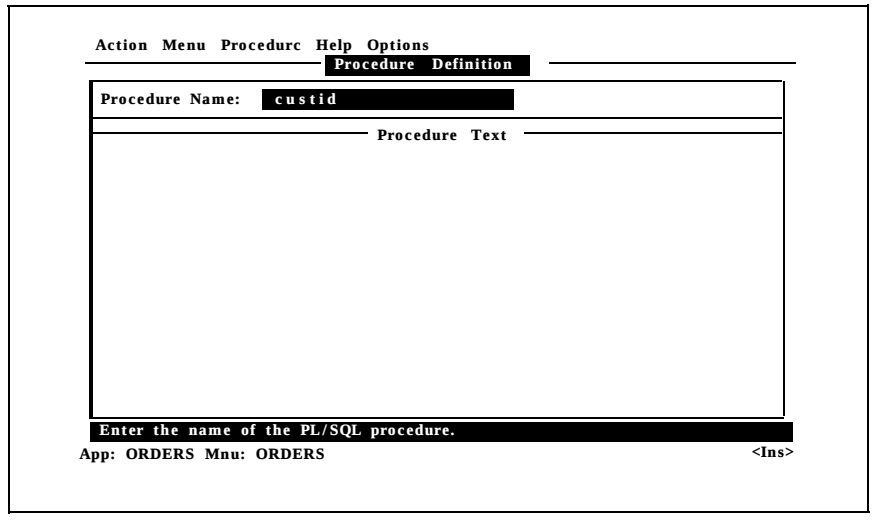
A *packaged procedure* is a built-in PL/SQL procedure. Each SQL*Menu packaged procedure performs a function, such as moving to the main menu or executing a background menu item.

Defining Procedures

You create and modify procedures in the SQL*Menu (Design) interface through the Procedure Definition form or spread table, as described in "Defining a PL/SQL Procedure" in Chapter 5.

The Procedure Definition form shows information for one procedure at a time (Figure 8-9), and the Procedure Definition spread table shows information for several procedures at once (Figure 8-10).

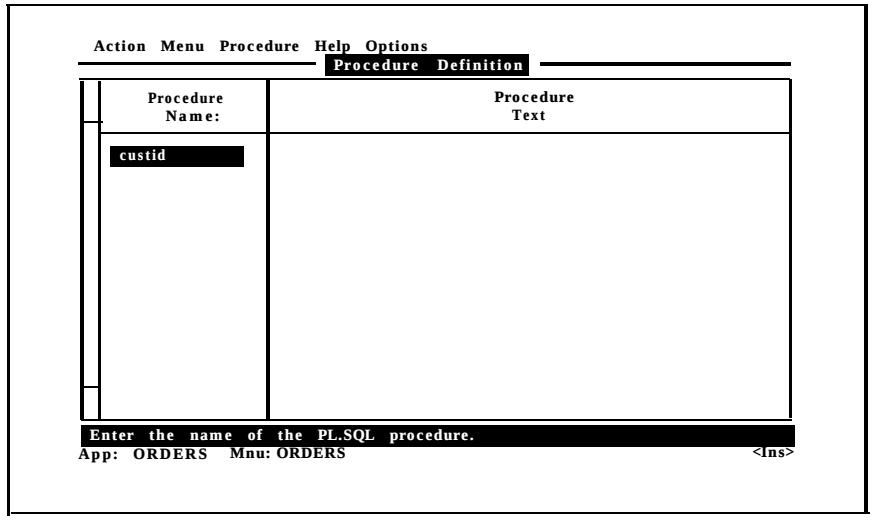
FIGURE 8-9
Procedure Definition Form



Location: To reach the Procedure Definition form or spread table from the main menu, complete the following steps:

1. Make sure you have application context. (You can also have menu and item context, but it is not necessary.)
2. Select the Procedure item from the main menu of the design interface.
3. Press [Change Display Type] to toggle the form and spread table display types.

FIGURE 8-10
Procedure Definition Spread Table



You can also display the Procedure Definition form or spread table by pressing [Navigate] from any form or spread table, or from any menu of the design interface, then specifying a procedure context

Procedure Characteristics

The process of defining a procedure consists of setting each of the procedure's characteristics. These characteristics indicate the following things about a procedure:

- procedure name
- procedure text

This section presents the procedure characteristics. The names of procedure characteristics correspond exactly with the names of the elements in the Procedure Definition form and the Procedure Definition spread table.

You assign procedure characteristics by typing values into field and scroll region elements.

Procedure Name

The Procedure Name characteristic specifies a name of up to 30 characters for the procedure. This name must satisfy the ORACLE naming conventions, and must match the name defined in the Procedure Text characteristic.

Required/Optional: required

Procedure Text

The Procedure Text characteristic specifies the PL/SQL block for the procedure, as described in "PL/SQL Blocks in SQL*Menu" in Chapter 9. You assign the Procedure Text characteristic in a scroll region in the Procedure Definition form or spread table.

Required/Optional: required

Using Procedures

Designers can use packaged procedures to perform the following tasks:

- reduce the amount of repetitive or complex data entry by operators
- control the flow of an application by forcing the operator to navigate along a predetermined path
- modify the default processing for processes such as validation and locking

You can only use a packaged procedure in the appropriate context, as described in "SQL*Menu States" in Chapter 10.

You can create your own procedures by using the Procedure Definition form or spread table, as described in “Defining a PL/SQL Procedure” in Chapter 5.

Note: The macros provided by previous versions of SQL*Menu (and still available in Version 5.0) are generally equivalent to PL/SQL packaged procedures, but the macro command type is less powerful than the PL/SQL command type, which can include packaged procedures, customized procedures, SQL statements, and other PL/SQL commands.

For more information about PL/SQL and packaged procedures, see Chapter 9, "PL/SQL," and Chapter 10, "PL/SQL Packaged Procedures.”

CHAPTER

9 PL/SQL

This chapter provides technical information about the PL/SQL language, including the following topics:

- PL/SQL overview
- PL/SQL blocks in SQL*Menu
- referencing objects in PL/SQL
- PL/SQL compilation
- executable statement failure

For information about PL/SQL packaged procedures, see Chapter 10, "PL/SQL Packaged Procedures."

This chapter does not discuss how to write PL/SQL programs. For this information, see the *PL/SQL User's Guide and Reference.*

PL/SQL Overview

PL/SQL is a database transaction processing language and an application development language. It combines extensive procedural capabilities with the nonprocedural capabilities of SQL.

PL/SQL supports the following features of SQL:

- **SQL data manipulation:** You can use INSERT, SELECT, UPDATE, and DELETE statements in PL/SQL.
- **SQL transaction processing:** You can use COMMIT, ROLLBACK, and SAVEPOINT statements in PL/SQL.
- **SQL functions:** You can invoke all SQL functions from within SQL statements in PL/SQL. PL/SQL statements that do not contain SQL coremands ("non-SQL statements") can call most SQL functions, but not the group functions.
- **SQL predicates:** You can use all the SQL predicates (WHERE clause conditions), including comparison operators and logical operators, in PL/SQL.

PL/SQL provides the following extensions to SQL:

- **Variables and constants:** You can use PL/SQL variables and constants anywhere an expression can be used, either in SQL statements or in non-SQL PL/SQL statements.
- **Logical comparisons:** PL/SQL supports the comparison of variables and constants by Boolean expressions, in SQL statements and non-SQL PL/SQL statements, for conditional control of block processing. PL/SQL can compare numeric, character, or date types of data.
- **Conditional control:** The IF statement controls whether or not a sequence of statements executes, using the clauses IF, THEN, ELSIF, and ELSE.
- **Iteractive control:** LOOP statements repeat a sequence of statements within a PL/SQL block. GOTO statements change the flow of control by branching within a PL/SQL block.
- **Cursor management:** APL/SQL *cursor* identifies the current row of data in a set of rows returned by a SQL query. You can use cursor attributes to access information about the execution of a multi-row query.
- **Error recovery:** PL/SQL's exception-handling mechanism can process errors that arise during-execution of a PL/SQL block. PL/SQL can handle both internally-defined and user-defined error conditions *(exceptions).*

For complete information about PL/SQL, see the *PL/SQL User's Guide and Reference.* For information about SQL, see the *SQL Language Reference Manual.*

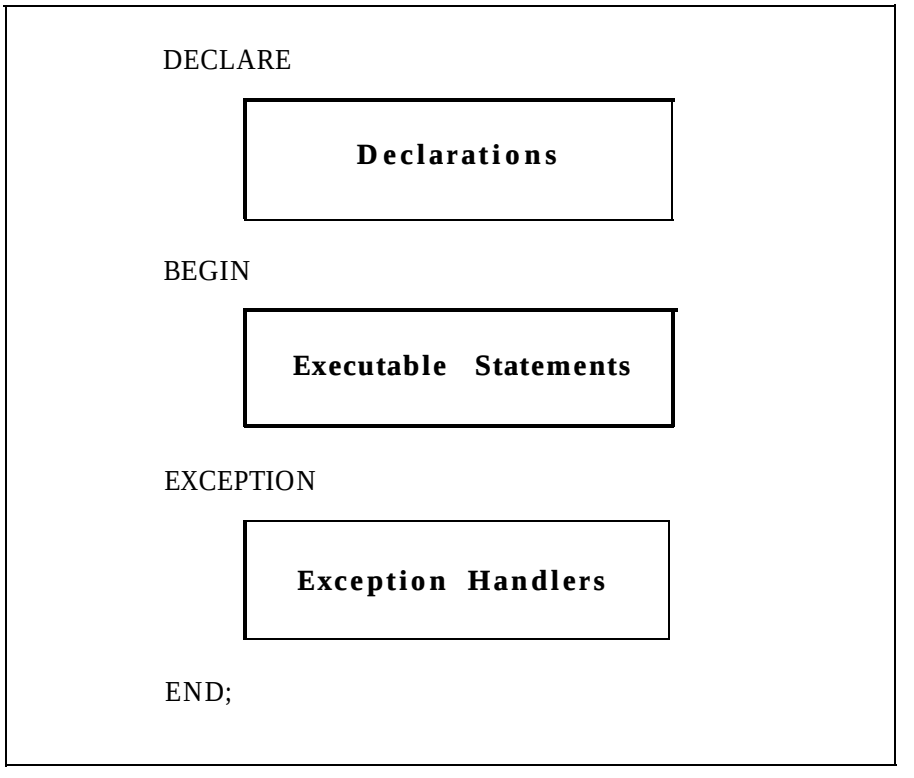
PL/SQL Blocks in SQL*Menu

A standard PL/SQL code segment is called a *block.* A block can consist of three parts, or sections:

- an optional declaration section for variables, constants, and exceptions
- a required section for executable statements
- an optional section for exception handlers

Figure 9-1 illustrates the order in which these sections can appear in a block and the PL/SQL keywords associated with a standard PL/SQL block.

FIGURE 9-1
Parts of a PL/SQL Block



SQL*Menu uses PL/SQL blocks in two different constructs:

- anonymous blocks
- procedure blocks

(procedures are referred to as form-level procedures in SQL*Forms.)

Anonymous Blocks

An anonymous block is a PL/SQL block that, unlike a procedure, has no name. This means that you can only execute an anonymous block from the menu item in which it is defined.

In SQL*Menu, an anonymous block does not usually require the explicit presence of the BEGIN and END keywords to enclose the executable statements. SQL*Menu automatically packages the BEGIN and END keywords with the block for compilation and execution.

If the anonymous block includes a declaration section, however, you do have to state BEGIN and END explicitly to separate the declaration section from the executable statements. An anonymous block requires a declaration section if the block uses local variables, PL/SQL cursors, or named exception handlers.

Note: You *can* always state the BEGIN and END keywords explicitly in an anonymous block. This practice affords syntactic consistency.

To define an anonymous block for a specific menu item, enter the PL/SQL block directly into the Command Line scroll region of the Item Definition form or spread table.

Procedure Blocks

A procedure is a PL/SQL block that requires a name. The full PL/SQL syntax, including declarations and the BEGIN and END keywords, is also required. Procedures let you create generic routines and use them in multiple places throughout an application. Procedures can also take arguments, while anonymous blocks cannot.

To define a PL/SQL procedure in SQL*Menu, enter the procedure name and the procedure text in the Procedure Definition form or spread table. You can then invoke this procedure in any PL/SQL block in the current menu application.

Procedures in SQL*Menu must contain PL/SQL procedure definitions with valid syntax. (They cannot contain an anonymous block.) Use the following syntax to define a PL/SQL procedure in the Procedure Text scroll region of the design interface:

Syntax

```
PROCEDURE procedure_name [argument_list] IS
    [local_variable_declaration]
BEGIN
    procedure_statements
    [EXCEPTION exception_handlers]
END;
```

where

procedure_name	Is the unique name of the procedure.
argument_list	Is ({ *var_name [mode] type* [:= *value*]} [, ...]) where: *var_name* is the unique name of a local variable. *mode* is [IN | 0UT | In OUT]. *type* is {BOOLEAN | CHAR | NUMBER | DATE}. *value* is a PL/SQL expression.
local_variable_declaration	Is { *var_name type* [:= *value*]} [, ...].
procedure_statements	Are the executable PL/SQL statements.
exception_handlers	(See the *PL/SQL User's Guide and Reference.)*

The name of a procedure must adhere to ORACLE naming conventions and be unique within a menu application. You must also ensure that this name matches the name of the procedure actually defined in the corresponding procedure text.

In the design interface you can copy or reference procedures that are defined in other menu applications by pressing [Copy Object] and using the Copy/Reference Object dialog box.

Referencing Objects in PL/SQL

In a PL/SQL procedure or anonymous block, you refer to substitution parameters and SQL*Forms global variables by the object names with which they are defined in the design interface or by their legal variable names.

Reference Syntax

The syntax you use to reference a parameter or variable object depends on whether you are referencing the value or the name of the object.

To reference the value of a substitution parameter or SQL*Forms global variable, prefix that object name with a colon:

```
: substitution_parameter
:GLOBAL. variablename  (for SQL*Forms variables)
```

To reference the name of an object, enclose the object name in single quotes without a colon prefix:

```
'menuname'
'itemname'
'fieldname'  or 'block.fieldname'  (for SQL*Forms objects)
```

Invoking Packaged Procedures

APL/SQL block can invoke any packaged procedure as an executable statement. To invoke a packaged procedure in a PL/SQL block, use the syntax specified for that packaged procedure in "SQL*Menu Packaged Procedures" in Chapter 10.

PL/SQL Variables

This section describes the local, global, and system variables available in PL/SQL.

Local Variables

A local variable is a PL/SQL variable that is active only within the anonymous block or procedure in which you declare it.

See the *PL/SQL User's Guide and Reference* for information on how to declare and reference local variables.

Global Variables

A global variable stores a character string value of any length. A global variable is active in any context within a menu application and is active throughout a SQL*Menu session. Because of this, a global variable declared in one menu can be used by any menus and forms that are called during the session.

Note: Global variables are only available in SQL*Menu for SQL*Forms variables, if menu and form applications are integrated.

Before a global variable can be active during a SQL*Menu (Run Menu) session, it must be initialized. SQL*Menu initializes a global variable the first time you assign a value to it.

You can delete an active global variable (and release the memory associated with it) with the ERASE packaged procedure.

Syntax `GLOBAL. variablename`

where *variablename* is a name that follows ORACLE naming conventions.

How to Assign Assign a value to a global variable with standard PL/SQL syntax for SQL*Menu objects:

```
:GLOBAL. variablename := value;
```

Common Usage Use global variables to store data values that you want to share among menus and forms during a SQL*Menu (Run Menu) session.

System Variables

A system variable keeps track of some internal state of the program. In general, you can reference the value of a system variable in order to control the way an application behaves.

Note: System variables are only useful in SQL*Menu when the menu application is invoked by a SQL*Forms application.

SQL*Menu maintains the values of system variables on a "per form" basis.

The following list presents the names of the available system variables, which are described in the *SQL*Forms Designer's Reference:*

SYSTEM.BLOCK_STATUS
SYSTEM.CURRENT _BLOCK
SYSTEM.CURRENT_FIELD
SYSTEM.CURRENT_FORM
SYSTEM.CURRENT_VALUE
SYSTEM.CURSOR_BLOCK
SYSTEM.CURSOR_FIELD
SYSTEM.CURSOR_RECORD
SYSTEM.CURSOR_VALUE
SYSTEM.FORM_STATUS
SYSTEM.LAST_QUERY
SYSTEM.LAST_RECORD
SYSTEM.MESSAGE_LEVEL
SYSTEM.RECORD_STATUS
SYSTEM.TRIGGER_BLOCK
SYSTEM.TRIGGER_FIELD
SYSTEM.TRIGGER_RECORD

Restrictions on Referencing Objects

The following restrictions apply to PL/SQL blocks that refer to SQL*Menu and SQL*Forms objects or functions:

- You can refer to SQL*Menu or SQL*Forms objects in PL/SQL blocks only when running PL/SQL within SQL*Menu or SQL*Forms.

  When you attempt to execute or compile a PL/SQL block from within the ORACLE RDBMS, you will receive errors if the block references any SQL*Menu or SQL*Forms objects.

- You can use packaged procedures in PL/SQL blocks only when running PL/SQL within SQL*Menu or SQL*Forms.
- You cannot directly refer to the value of a SQL*Forms object. You can only refer to the object's name using the SQL*Forms NAME_IN packaged function.
- You cannot directly assign a value to a SQL*Forms object. For example, you cannot assign the value SALESMAN to the JOB field with the statement : `JOB := 'SALESMAN'`.

  You can use the SQL*Forms COPY packaged procedure to assign a value to a SQL*Forms object, such as `COPY('SALESMAN', 'JOB');`.

PL/SQL Compilation

PL/SQL blocks need to be compiled before they can be executed. When a PL/SQL block is compiled, the compiler detects syntax errors, semantic errors, and references to non-existent objects or procedures.

When you use anonymous blocks and procedures, SQL*Menu compiles them immediately after you create or modify them *and* when you generate a menu application that contains them.

Note: SQL*Menu discards all of an application's existing compilation information, retaining only PL/SQL source, when you save the application or before you generate a DMM library file.

Compiling at Design Time

SQL*Menu compiles PL/SQL procedures and anonymous blocks as soon as you create them in the SQL*Menu (Design) interface:

- SQL*Menu compiles a PL/SQL anonymous block in a menu item's command line after you write or modify the block and press [Accept].
- SQL*Menu compiles a procedure after you write or modify the procedure in the Procedure Definition form or spread table and press [Accept].

If the compiler identifies any errors in a procedure or an anonymous block, SQL*Menu displays the Compilation Errors form. This form highlights the first error in the procedure or block and displays explanatory text for all of the errors found during compilation.

Compiling at Generation Time

SQL*Menu compiles all of the PL/SQL blocks and procedures in a menu application when you generate an application library file.

The *procedure specification* is the part of the procedure text that includes the procedure name and any associated argument list. SQL*Menu compiles this information before the executable statements, to verify that the procedure exists and that it can be referenced as specified.

When you generate an application, SQL*Menu compiles the following constructs in the following order:

1. SQL*Menu scans all procedures and compiles each procedure specification.
2. SQL*Menu compiles each procedure.
3. SQL*Menu identifies all menu items of command type 7 and compiles each anonymous block.

Compilation Error Messages

As SQL*Menu compiles each procedure and anonymous block, it generates messages identifying all procedures and menu items of command type 7 and reporting any errors and error locations.

If you generate a menu application from the command line, SQL*Menu displays the messages on your terminal screen, one at a time. If you generate a menu application from SQL*Menu (Design) (i.e., via the Generate option on the Action submenu), SQL*Menu writes the messages to a file in your default directory. This file is named *application_file_name*`.err`, where *application_file_name* is the name of the current menu application.

The compilation error messages take the following form:

```
Compiling object_name object_type ...
Error error_num at line line_num, column col_num (sev n) :
        error_text
```

where:

object_name	Is the unambiguous name of the procedure or block that SQL*Menu is compiling. For a menu item, *object name* is the value of the Short Item Name field in the Item Definition form or spread table.
object_type	Is `item`, `procedure`, or `procedure specification`.
error_num	Is the three-digit PL/SQL error number documented in the *ORACLE Error Messages and Codes Manual*.
line_num	Is the line number of the procedure or block where the error occurred. This is the line number within the Item Text scroll region or the Procedure Text scroll region.
col_num	Is the column number of the procedure or block where the error occurred. This is the column number within the Item Text scroll region or the Procedure Text scroll region.
`(sev n)`	Is the severity of the error condition, where *n* is an integer.
error_text	Is a complete description of the error.

Executable Statement Failure

In PL/SQL blocks you trap, or isolate, success and failure differently depending on whether the executable statement is a SQL statement or a packaged procedure. Use the appropriate error-processing method when the success or failure of an executable statement is important to the execution of your application.

Packaged procedure failure or success can be trapped by the following PL/SQL packaged functions:

MEM_SUCCESS	FORM_SUCCESS	ERROR_TYPE
MENU_FAILURE	FORM_FAILURE	ERROR_CODE
MENU_FATAL	FORM_FATAL	ERROR_TEXT

These functions are discussed in Chapter 10, “PL/SQL Packaged Procedures.”

SQL Statement Failure

Use exception handling to trap the successor failure of a SQL statement. The PL/SQL functions SQLCODE and SQLERRM are available to identify specific errors. Refer to the *PL/SQL User's Guide and Reference* for information on how to use these functions.

Exception Handling

Exception handlers define statements that are to be executed when certain conditions, or exceptions, are raised in a PL/SQL block. Some exceptions are predefine and are raised internally by PL/SQL (for example, ZERO_DIVIDE), while other exception handlers are user-defined and must be raised explicitly with the RAISE statement.

When an exception is raised, PL/SQL stops the normal execution of the PL/SQL block and passes control to the appropriate exception handler. The execution of a PL/SQL block stops when the exception handler completes execution.

Refer to the *PL/SQL User's Guide and Reference* for an explanation of exception handling.

CHAPTER

10

PL/SQL PACKAGED PROCEDURES

This chapter provides technical information about PL/SQL packaged procedures, including the following topics

- packaged procedure overview
- packaged procedure failure
- packaged procedure arguments
- SQL*Menu packaged procedures
- SQL*Forms packaged procedures

Packaged Procedure Overview

A packaged procedure is a named PL/SQL command that allows designers to specify functions normally performed by function keys. You can invoke packaged procedures in a menu item's command line or in a PL/SQL procedure.

Most SQL*Menu packaged procedures have a corresponding macro command. The packaged procedure QUERY_PARAMETER is exceptional because it performs a function that is only necessary in PL/SQL.

Some SQL*Menu (Run Menu) functions can only be executed with function keys. These functions are not practical for execution in a packaged procedure because they are often performed by the terminal or user interface management system and not by SQL*Menu.

For a list of the function keys and macros that correspond to each packaged procedure, see "Macros, Packaged Procedures, and Function Keys" in Chapter 11.

Packaged Procedure Failure

A packaged procedure can fail for two reasons:

- It is not allowed in the current state of SQL*Menu.
- It could not execute correctly.

The following sections describe the SQL*Menu states that can affect packaged procedure execution and the PL/SQL functions that you can use to test the successor failure of packaged procedures.

The definitions in "SQL*Menu Packaged Procedures" later in this chapter list the valid states of SQL*Menu for each packaged procedure, and also describe the conditions that can cause failure when a packaged procedure executes in a valid state.

SQL*Menu States

Certain functions have no meaning in some menus or states of SQL*Menu. Similarly, their corresponding packaged procedures have no meaning in the same states.

SQL*Menu packaged procedures can execute in the following states of SQL*Menu (Run Menu):

LOGIN	Login state. SQL*Menu is in the LOGIN state when it tries to log into ORACLE. Once SQL*Menu has logged in, the state changes to AMENU or MENU, if an application was specified.
AMENU	Application Menu state. SQL*Menu is in the AMENU state if the current menu is the Application Menu. This state changes to MENU when an application is selected.
MENU	Menu state. SQL*Menu is in the MENU state if a menu of an application is active. Functions that occur infields (such as MENU_CLEAR_FIELD) are only valid in the MENU state for applications running in the full-screen menu style. Depending on what function is executed, the MENU state can change to any of several states: AMENU,MENU AFORM, MFORM, or IFORM.
AFORM	Application Parameter Form state. SQL*Menu is in the AFORM state if the Enter Parameter Values dialog box is displayed for application parameters. This state changes to MENU after exiting the dialog box.
MFORM	Menu Parameter Form state. SQL*Menu is in the MFORM state if the Enter Parameter Values dialog box is displayed for menu parameters in a full-screen menu. This state can change to MENU or IFORM.
IFORM	Item Parameter Form state. SQL*Menu is in the IFORM state if the Enter Parameter Values dialog box is displayed for item parameters. This state changes to MENU after the item is executed, or if the dialog box is aborted by the [Cancel] function.

Testing Packaged Procedure Execution

SQL*Menu provides the following PL/SQL functions that you can use to trap the successor failure of packaged procedures:

MENU_SUCCESS	FORM_SUCCESS	ERROR_TYPE
MENU_FAILURE	FORM_FAILURE	ERROR_CODE
MENU_FATAL	FORM_FATAL	ERROR_TEXT

Each of these functions tests the value of a specific error variable that is set whenever SQL*Menu executes a packaged procedure.

The error variables record the following information:

- whether the most recent statement succeeded, failed, or caused a fatal error
- the type, number, and text of any associated error message

Note: If your application is supported in multiple languages, avoid testing with the ERROR_TEXT packaged function. Instead, use the ERROR_TYPE and ERROR_CODE functions to test for a specific error.

Because each packaged procedure sets the return values of these functions when it executes, the functions only report information on the most recently executed packaged procedure. Other actions such as pressing a function key can also set these return values, so you should always test the results immediately after a packaged procedure executes.

You can alter the flow of processing by testing the value of any of the error variables through the appropriate PL/SQL function, and then raising an exception or taking the appropriate action.

Packaged Procedure Arguments

Some packaged procedures require string or number arguments. For conciseness, the syntax for packaged procedures represents arguments with names that describe what the argument must evaluate to. For example, the NEW_USER packaged procedure has an argument called *login_string*. This syntax indicates that the NEW_USER packaged procedure requires a character-string argument that evaluates to the login ID for a user.

There are many different ways to construct a valid character string or number argument. The following information details the types of expressions that you can use for packaged procedure arguments that must evaluate to character strings or numbers.

literal reference — *'argument_ string** or *number,* where

argument_string is the character string that satisfies the argument.

number is the integer that satisfies the argument.

PL/SQL variable reference

variable_name

where *variable_name* is the name of a PL/SQL local variable whose value is the character string or integer that satisfies the argument. In SQL*Menu, the named variable is a substitution parameter. See the *PL/SQL User's Guide and Reference* for information on PL/SQL local variables.

indirect reference

```
NAME_IN (argument_reference)
```

where *argument_reference* can be any kind of argument reference (including another indirect reference) that evaluates to the name of a field or variable whose value is the character string or integer that satisfies the argument. Use indirect reference only to refer to SQL*Forms variables. See the *SQL*Forms Designer's Reference* for information on the NAME_IN packaged function.

Arguments can also be constructed with combinations of these expressions. You can also use other general PL/SQL expressions such as concatenation for character-string arguments and addition for number arguments.

SQL*Menu Packaged Procedures

This section describes each packaged procedure with its syntax, the equivalent macro, and the SQL*Menu states in which it is valid. Failure conditions are also specified.

APPLICATION_MENU

Syntax:

```
APPLICATION_MENU;
```

Equivalent Macro: APLMENU

Description: APPLICATION_MENU navigates to the application menu from a SQL*Menu application. This packaged procedure is not allowed if SQL*Forms invoked SQL*Menu.

Valid States: MENU, AFORM, MFORM, IFORM

APPLICATION_PARAMETER

Syntax: `APPLICATION_PARAMETER;`

Equivalent Macro: APLPARM

Description: APPLICATION_PARAMETER displays all the parameters associated with the current application, and their current values, in the Enter Parameter Values dialog box.

Valid States: MENU, MFORM

Failure: If no parameters are defined for the current application, you will get error message `MNU-10201: No parameters needed.`

BACKGROUND_MENU*n*

Syntax: `BACKGROUND_MENU {l |2 |3 |4 |5 |6 |7 |8 |9 |10};`

Equivalent Macros: BGM1 through BGM10

Description: BACKGROUND_MENU *n* executes the designer-specified item *n* from the background menu, as described in "Using the Background Menu" in Chapter 2.

Valid States: MENU, AFORM, MFORM, IFORM

Failure: If background menu item *n* does not exist for the current application, you will get error message `MNU -10204:NO command defined for the selected background item.` If background menu item *n* tries to navigate to a menu, you will get error message `MNU-102 09:No next menu' from background in this context.`

DEBUG_MODE

Syntax: `DEBUG_MODE;`

Equivalent Macro: DEBUG

Description: DEBUG_MODE toggles the debug mode on and off, as described in "Executing a Menu Application" in Chapter 5.

Valid States: MENU

DISABLE_ITEM

Syntax: `DISABLE_ITEM (menu_name, item_short_name);`

Equivalent Macro: (none)

Description: DISABLE_ITEM makes the indicated menu item inaccessible to users. DISABLE_ITEM displays the item in grey characters, instead of bold characters, even if the Display without Privilege characteristic is turned off for the item.

DISABLE_ITEM can override a user's access privilege to the specified item. For example, you can disable a menu item for a user who has access privilege to the item.

Note: You can use DISABLE_ITEM in SQL*Menu commands *and* in SQL*Forms triggers and form-level procedures. This capability, which is unique to DISABLE_ITEM and ENABLE_ITEM among SQL*Menu packaged procedures, lets you disable menu items directly from within a form.

Valid States: MENU, AFORM, MFORM, IFORM

Identifying Disabled Menu Items

You can determine if a menu item is enabled or disabled by using the ITEM_ENABLED packaged function, with the following syntax:

```
ITEM_ENABLED (menu_name, item_short_name) ;
```

This Boolean function returns TRUE if the specified item is enabled and FALSE if it is disabled or non-existent.

Note: You can use ITEM_ENABLED in SQL*Menu commands *and* in SQL*Forms triggers and form-level procedures. This capability, which is unique to ITEM_ENABLED among SQL*Menu packaged functions, lets you determine the status of menu items directly from within a form.

ENABLE_ITEM

Syntax: `ENABLE_ITEM (menu_name,item_short_name);`

Equivalent Macro: (none)

Description: ENABLE_ITEM makes the indicated menu item accessible to users and displays the item in bold characters, instead of grey characters.

ENABLE_ITEM can override a user's access privilege to the specified item. For example, you can enable a menu item for a user who does not have access privilege to the item. However, you can only enable an inaccessible menu item that is currently visible (in grey characters) to the user.

Note: You can use ENABLE_ITEM in SQL*Menu commands *and* in SQL*Forms triggers and form-level procedures. This capability, which is unique to ENABLE_ITEM and DISABLE_ITEM among SQL*Menu packaged procedures, lets you enable menu items directly from within a form.

Valid States: MENU, AFORM, MFORM, IFORM

Identifying Enabled Menu Items

You can determine if a menu item is enabled or disabled by using the ITEM_ENABLED packaged function, with the following syntax:

```
ITEM_ENABLED (menu_name, item_short_name) ;
```

This Boolean function returns TRUE if the specified item is enabled and FALSE if it is disabled or non-existent.

Note: You can use ITEM_ENABLED in SQL*Menu commands *and* in SQL*Forms triggers and form-level procedures. This capability, which is unique to ITEM_ENABLED among SQL*Menu packaged functions, lets you determine the status of menu items directly from within a form.

EXIT_MENU

Syntax: `EXIT_MENU;` or `EXIT_MENU` *`(message_string);`*

Equivalent Macro: EXIT

Description: EXIT_MENU navigates to "outside" of the menu. The message text appears on the screen (for example, `EXIT_MENU ('Menu terminated');`).

Valid States: LOGIN, AMENU, MENU, AFORM, MFORM, IFORM

MAIN_MENU

Syntax: `MAIN_MENU;`

Equivalent Macro: MAINMENU

Description: MAIN_MENU navigates to the main menu of the current application.

Valid States: MENU, AFORM, MFORM, IFORM

Failure: If SQL*Menu is already in the main menu, you will get error message `MNU-10200: Illegal function in this context .`

MENU_CLEAR_FIELD

Syntax: `MENU_CLEAR_FIELD;`

Equivalent Macro: CLRFLD

Description: MENU_CLEAR_FIELD clears the current field's value from the current cursor position to the end of the field. If the current cursor position is to the right of the last nonblank character, MENU_CLEAR_FIELD clears the entire field, making its value NULL.

Valid States: LOGLN, AMENU, MENU, AFORM, MFORM, IFORM

MENU_HELP

Syntax: `MENU_HELP;`

Equivalent Macro: HELP

Description: MENU_HELP displays the current menu item's hint message on the message line.

Valid States: LOGIN, AMENU, MENU, AFORM, MFORM, IFORM

Failure: If no help information is available for the current menu item, you will get error message `MNU-10215: No help available`.

MENU_MESSAGE

Syntax: `MENU_MESSAGE` *`(message_string);`*

Equivalent Macro: (none)

Description: MENU_MESSAGE displays the specified message on the message line.

Valid States: LOGIN, AMENU, MENU, AFORM, MFORM, IFORM

MENU_NEXT_FIELD

Syntax: `MENU_NEXT_FIELD;`

Equivalent Macro: NXTFLD

Description: MENU_NEXT_FIELD navigates to the next field in a SQL*Menu form or dialog box.

Valid States: LOGIN, AMENU, MENU, AFORM, MFORM, IFORM

MENU_PARAMETER

Syntax: `MENU_PARAMETER;`

Equivalent Macro: MENUPARM

Description: MENU_PARAMETER displays all the parameters associated with the current menu, and their current values, in the Enter Parameter Values dialog box.

Valid States: MENU, MFORM (in full-screen display style)

Failure: If no menu parameters are defined for the current menu, you will get error message `MNU-10201: No parameters needed.`

MENU_PREVIOUS_FIELD

Syntax: `MENU_PREVIOUS_FIELD;`

Equivalent Macro: PRVFLD

Description: MENU_PREVIOUS_FIELD returns to the previous field in a SQL*Menu form or dialog box.

Valid States: LOGIN, AFORM, MFORM, IFORM

MENU_REDISPLAY

Syntax: `MENU_REDISPLAY;`

Equivalent Macro: REDISP

Description: MENU_REDISPLAY redraws the screen in SQL*Menu (Run Menu). This eliminates any extraneous system messages.

Valid States: LOGIN, AMENU, MENU, AFORM, MFORM, IFORM

MENU_SHOW_KEYS

Syntax: `MENU_SHOW_KEYS;`

Equivalent Macro: SHOWKEYS

Description: MENU_SHOW_KEYS displays the Show Keys screen for SQL*Menu (Run Menu).

Valid States: LOGIN, AMENU, MENU, AFORM, MFORM, IFORM

NEW_APPLICATION

Syntax: `NEW_APPLICATION` *`(application_name);`*

Equivalent Macro: NEWAPL

Description: NEW_APPLICATION changes the current menu application.

Valid States: AMENU, MENU, AFORM, MFORM, IFORM

Failure: If the specified application does not exist in the database, or if you do not have access privilege for the application, you will get error message `MNU-l0244: Application` *`name`* `does not exist.` This packaged procedure is not allowed in SQL*Forms.

NEW_USER

Syntax: `NEW_USER` *`(login_string);`*

Equivalent Macro: NEWUSER

Description: NEW_USER logs out the current user and logs in the user identified by *`login_string`*. For example, if *`login_string`* is SCOTT/TIGER, the command syntax is `NEW_USER ('SCOTT/TIGER');`. This packaged procedure is not allowed in SQL*Forms.

Valid States: AMENU, MENU, AFORM, MFORM, IFORM

NEXT_MENU_ITEM

Syntax: `NEXT_MENU_ITEM;`

Equivalent Macro: DOWN

Description: NEXT_MENU_ITEM navigates to the next menu item in the current menu.

Valid States: AMENU, MENU

OS_COMMAND

Syntax: `OS_COMMAND (command_string);`

Equivalent Macro: OSCMD

Description: OS_COMMAND executes the operating system command specific in *command_string*. For example, `OS_COMMAND ('(TYPE=3)DIR');` executes the command `DIR` as a type 3 command (after execution pause for operator response). The command type can be 2, 3, 4, or 5, and the default is type 2 (return immediately after execution).

Valid States: MENU, AFORM, MFORM, IFORM

OS_COMMAND1

Syntax: `OS_COMMAND1 (command_string);`

Equivalent Macro: OSCMD1

Description: OS_COMMAND1 executes the operating system command specified in *command_string*. For example, `OS_COMMAND1('(TYPE=3)DIR');` executes the command `DIR` as a type 3 command (after execution pause for operator response). The command type can be 2, 3, 4, or 5, and the default is type 2 (return immediately after execution).

Valid States: MENU, AFORM, MFORM, IFORM

PREVIOUS_MENU

Syntax: `PREVIOUS_MENU;`

Equivalent Macro: PRVMENU

Description: PREVIOUS_MENU returns to the previous menu.

Valid States: MENU, AFORM, MFORM, IFORM

PREVIOUS_MENU_ITEM

Syntax: `PREVIOUS_MENU_ITEM;`

Equivalent Macro: UP

Description: PREVIOUS_MENU_ITEM navigates to the previous menu item in the current menu.

Valid States: AMENU, MENU

QUERY_PARAMETER

Syntax: `QUERY_PARAMETER` *`(parameter_string);`*

Equivalent Macro: (none)

Description: QUERY_PARAMETER queries SQL*Menu for the current values of a list of substitution parameters, so that PL/SQL can use the values in processing commands. The parameter names in the QUERY_PARAMETER argument must have an ampersand prefix *`('&P1 &p2 &p3 ... '),`* although a colon prefix *`(:p1)`* appears in the PL/SQL code that uses the parameter values.

Valid States: MENU, AFORM, MFORM, IFORM

SHOW_BACKGROUND_MENU

Syntax: `SHOW_ BACKGROUND MENU;`

Equivalent Macro: SHOWBGM

Description: SHOW_BACKGROUND_MENU displays the background menu, as described in "Using the Background Menu" in Chapter 2.

Valid States: MENU, AFORM, MFORM, IFORM

Failure: If no background menu is defined for the current application, you will get error messge `MNU-10207: No background menu present.`

TERMINATE

Syntax: `TERMINATE;`

Equivalent Macro: TRMNATE

Description: TERMINATE terminates input in a form or dialog box. This function is equivalent to the operator pressing [Accept].

Valid States: LOGIN, AMENU, MENU, AFORM, MFORM, IFORM

WHERE_DISPLAY

Syntax: `WHERE_DISPLAY;`

Equivalent Macro: WHERE

Description: WHERE_DISPLAY toggles the Where option navigation display on and off, as described in Chapter 3, "Menu Styles and Navigation."

Valid States: AMENU, MENU (only in full-screen display style)

SQL*Forms Packaged Procedures

When a SQL*Forms application invokes a SQL*Menu application, SQL*Menu can execute SQL*Forms packaged procedures from an item command line or a PL/SQL procedure, without restriction.

The following packaged procedures are available in SQL*Forms:

ABORT_QUERY	DOWN	NEXT_FIELD
ANCHOR_VIEW	DUPLICATE_FIELD	NEXT_KEY
BELL	DUPLICATE_RECORD	NEXT_RECORD
BLOCK_MENU	EDIT_FIELD	NEXT_SET
BREAK	ENTER	PASTE_REGION
CALL	ENTER_QUERY	PAUSE
CALL_INPUT	ERASE	POST
CALL_QUERY	EXECUTE_QUERY	PREVIOUS_BLOCK
CLEAR_BLOCK	EXECUTE_TRIGGER	PREVIOUS_FIELD
CLEAR_EOL	EXIT_FORM	PREVIOUS_RECORD
CLEAR_FIELD	FIRST_RECORD	PRINT
CLEAR_FORM	GO_BLOCK	REDISPLAY
CLEAR_RECORD	GO_FIELD	REPLACE_MENU
COMMIT_FORM	GO_RECORD	RESIZE_VIEW
COPY	HELP	SCROLL_DOWN
COPY_REGION	HIDE_MENU	SCROLL_UP
COUNT_QUERY	HIDE_PAGE	SET_FIELD
CREATE_RECORD	HOST	SET_INPUT_FOCUS
CUT_REGION	LAST_RECORD	SHOW_KEYS
DEFAULT_VALUE	LIST_VALUES	SHOW_MENU
DELETE_RECORD	LOCK_RECORD	SHOW_PAGE
DISPLAY_ERROR	MESSAGE	SYNCHRONIZE
DISPLAY_FIELD	MOVE_VIEW	UP
DISPLAY_PAGE	NEW_FORM	USER_EXIT
DO_KEY	NEXT_BLOCK	

The SQL*Forms packaged procedure REPLACE_MENU replaces the current menu with a menu from any menu application. Arguments to REPLACE_MENU can specify the menu display style and the role to use, as well as the application name and starting menu name.

SQL*Forms packaged procedures SHOW MENU and HIDE MENU display and conceal the current menu, respectively.

The SQL*Forms packaged procedure SET_INPUT_FOCUS “activates” a menu m a character mode environment. This packaged procedure sets the input focus of the application to the current menu.

See the *SQL*Forms Designer’s Reference* for descriptions of the SQL*Forms packaged procedures and their syntax.

CHAPTER

11

SQL*MENU MACROS

This chapter provides technical information about SQL*Menu macro commands, including the following topics:

- macro overview
- macro failure
- macros without packaged procedures
- macros with arguments

Macro Overview

Macro statements allow application designers to specify functions in a menu item's command line that would normally be performed by an operator pressing function keys.

In SQL*Menu Version 5, the macro functions can be specified by PL/SQL statements. Macros are available primarily to make menu applications created in SQL*Menu Version 4.1 upwardly compatible with SQL*Menu Version 5.0.

Macro Syntax

A macro statement can contain one macro or a string of several macros, in the following syntax:

```
macro1; [macro2; ...]
```

If a macro takes an argument, the argument follows the macro name:

```
macro1 arg1;
```

Macros, Packaged Procedures, and Function Keys

Most SQL*Menu macros have a corresponding packaged procedure, and vice versa. Table 11-1 shows the correspondence and lists the exceptions between macros and packaged procedures.

Some SQL*Menu (Run Menu) functions can only be executed with function keys. These functions are not practical for execution in a macro or procedure because they are often performed by the terminal or user interface management system and not by SQL*Menu.

Table 11-1 lists the function keys that correspond to each SQL*Menu macro and packaged procedure.

For descriptions of the macros that have corresponding packaged procedures, see the corresponding descriptions in "SQL*Menu Packaged Procedures" in Chapter 10.

TABLE 11-1
Macros, Packaged Procedures, and Function Keys

Macro Name	*Packaged Procedure*	*Function Key*
APLMENU	APPLICATION_MENU	[Application Menu]
APLPARM	APPLICATION_PARAMETER	[Enter Application Parameters]
ASSIGN	none	none
BGM *n*	BACKGROUND_MENU *n*	[Background Menu *n*]

Table continued on next page . . . .

TABLE 11-1
Macros, Packaged Procedures, and Function Keys

Macro Name	*Packaged Procedure*	*Function Key*
CHRMODE	none	[Insert/Replace]
CLRFLD	MEM_CLEAR_FIELD	[Clear Field]
DEBUG	DEBUG_MODE	[Debug Mode]
DELCHR	none	[Delete Backwards]
DISP	none	none
DOWN	NEXT_MENU_ITEM	[Down]
EXIT	EXIT_MENU	[Quit]
HELP	MENU_HELP	[Help]
LEFT	none	[Left]
MAINMENU	MAIN_MENU	[Main Menu]
MENUPARM	MENU_PARAMETER	[Enter Menu Parameters]
NEWAPL	NEW_APPLICATION	none
NEWUSER	NEW_USER	[Redefine Usr/Pwd]
NODISP	none	none
NXTFLD	MENU_NEXT_FIELD	[Next Field]
OSCMD	OS_COMMAND	[Enter >1 OS Command]
OSCMD1	OS_COMMANDl	[Enter 10S Command]
PRVFLD	MENU_PREVIOUS_FIELD	[Previous Field]
PRVMENU	PREVIOUS_MENU	[Previous Menu]
REDISP	MEM_REDISPLAY	[Refresh]
RIGHT	none	[Right]
SHOWBGM	SHOW_BACKGROUND_MENU	[Show BGMJ
SHOWKEYS	MENU_SHOW_KEYS	[Show Keys]
SUSPEND	none	none
TRMNATE	TERMINATE	[Accept]
UP	PREVIOUS_MENU_ITEM	[up]
WHERE	WHERE_DISPLAY	[Where Display]
none	DISABLE_ITEM	none
none	ENABLE_ITEM	none
none	MENU_MESSAGE	none
none	QUERY_PARAMETER	none

Macro Failure

If a macro statement fails, SQL*Menu stops processing the macro stream and returns control to the user at the point the failed macro stopped executing.

The error messages that can appear when a macro fails are explained in "SQL*Menu (Run Menu) Error Messages" in Appendix B (for example, messages MNU-10233 to MNU-10238 and MNU-10241).

A macro can fail for two reasons:

- It is not allowed in the current state of SQL*Menu.
- It could not execute correctly.

Certain functions have no meaning in some menus or states of SQL*Menu. Similarly, their corresponding macro functions have no meaning in the same states.

The following states of SQL*Menu (Run Menu) can affect macro processing:

- LOGIN state
- AMENU state (application menu)
- MENU state
- AFORM, MFORM, and IFORM states (Enter Parameter Values dialog box)

These states of SQL*Menu (Run Menu) are described in "SQL*Menu States" in Chapter 10.

Valid states for each macro that has a corresponding packaged procedure are listed in "SQL*Menu Packaged Procedures" in Chapter 10, which also lists conditions that can cause failure when a macro is executed in a valid SQL*Menu state.

The macros that do not have corresponding packaged procedures are valid in any SQL*Menu state. (Macros that are valid within a field-CHRMODE, DELCHR, LEFT, and RIGHT—function in the full-screen display style of the AMENU and MENU states.)

The ASSIGN macro fails to execute when it tries to assign a value to one of the predefined parameters (UN, PW, TT, SO, and AD), as described in "Macros with Arguments" later in this chapter.

Macros without Packaged Procedures

This section describes the SQL*Menu macro commands that do not have corresponding packaged procedures.

ASSIGN ASSIGN defines the value of a substitution parameter, as described in the following section, "Macros with Arguments."

CHRMODE CHRMODE toggles the insert/replace typing mode, changing the status line indicator from `<Ins>` to `<Rep>` or from `<Rep>` to `<Ins>`.

DELCHR DELCHR deletes the character to the left of the cursor.

DISP DISP turns on the screen update function in character-mode displays. By default, SQL*Menu suppresses output to the screen during the execution of a macro by adding `NODISP;` to the front of every macro statement and `DISP; REDISP;` to the end.

To make sure that the screen is updated after every macro function is executed, use the DISP macro followed by REDISP to refresh the screen. Then to suppress the screen output, use the NODISP macro.

LEFT LEFT moves the cursor one position to the left in a field.

NODISP NODISP turns off the screen update function for this macro in character-mode displays. By default, SQL*Menu suppresses output to the screen during the execution of a macro by adding `NODISP;` to the front of every macro statement and `DISP; REDISP;` to the end.

RIGHT RIGHT moves the cursor one position to the right in a field.

SUSPEND SUSPEND interrupts macro processing until the menu operator presses [Accept].

Macros with Arguments

This section describes the SQL*Menu macro commands that take arguments:

ASSIGN

The ASSIGN macro can assign values to parameters without using the Enter Parameter Values dialog box. This lets you execute menu items that have parameters without requiring operator input.

Syntax The syntax for the ASSIGN macro is as follows:

```
ASSIGN parameter=value;
```

where *parameter* is the two-letter parameter name and *value* is any string. Do not put spaces around the equals sign (=).

For example, the following command assigns the value "edit.txt" to the FN parameter:

```
ASSIGN FN=edit.txt;
```

Assigning a value to the predefine parameter TT, AD, UN, or PW results in an error, but you can assign a value to the predefine parameter SO which identifies the current menu item. This assignment should be followed by a TRMNATE; otherwise, a REDISP will cause the assignment to be lost. Since SQL*Menu performs an implicit REDISP at the end of every macro statement, assigning values to SO should be done with caution.

EXIT

The EXIT macro optionally can take a string as an argument. This string is printed as a message when SQL*Menu terminates.

Syntax The syntax for the EXIT macro is as follows:

```
EXIT text_ string;
```

For example, the following command prints the string "SQL*Menu Terminated" before stopping SQL*Menu:

```
EXIT SQL*Menu Terminated;
```

NEWAPL

The NEWAPL macro allows you to change applications within a macro.

Syntax The syntax for the NEWAPL macro is as follows:

```
NEWAPL application_name;
```

NEWUSER

The NEWUSER macro can be used to log in to ORACLE using a new user ID and password, or to log in to another database.

Syntax The syntax for the NEWUSER macro is as follows:

```
NEWUSER [login_string];
```

where *login_string=username/password[@database]*

Note: Do not enter a space before *@database*.

OSCMD and OSCMD1

Both OSCMD and OSCMD1 allow you to execute operating system commands within a macro.

Syntax The syntax for the OSCMD and OSCMD1 macros is as follows:

```
OSCMD [command];
OSCMD1 [command];
```

where `command = [(TYPE = 2 |3 |4 |5)]` *command_string*.

In cases where *command_string* begins with the character" (", you must specify a command type:

Type 2	Execute an operating system command and return immediately to SQL*Menu (Run Menu)
Type 3	Execute an operating system command and pause before returning to SQL*Menu (Run Menu)
Type 4	Invoke SQL*Forms (Run Form)
Type 5	Invoke SQL*Plus

The default is command type 2. These command types correspond to the types specified in the Item Definition form or spread table, as described in "SQL*Menu Command Types" in Chapter 8.

SQL*Menu passes the *command_string* to the operating system for execution.

For example, the following command runs the form named ORDERS, using SCOTT as the ORACLE username and TIGER as the password:

```
OSCMD1 (TYPE = 4) RUNFORM ORDERS SCOTT/TIGER;
```

Note: If SQL*Menu and SQL*Forms are linked, you must use command type 2 to specify SQL*Forms command line switches -c, -e, -i, -r, and -w.

CHAPTER

12

SQL*MENU COMPONENTS AND ADMINISTRATION

This chapter provides reference information on the following topics for SQL*Menu administrators and designers

- automatic login
- remote login
- SQL*Menu system components
- SQL*Menu constraints
- terminal definitions
- views of the SQL*Menu base tables

Automatic Login

The ORACLE automatic login feature allows you to log into ORACLE products without explicitly stating a username and password. When you log in in this way, ORACLE uses an OPS$ ID to associate your ORACLE username to your operating system User ID.

You can use the automatic login feature with all SQL*Menu components. For example, to log in to SQL*Menu (Run Menu) using your default OPS$ login, enter `RUNMENU/` at the operating system prompt. SQL*Menu logs you in, bypassing the login screen.

If your SQL*Menu applications use the &UN and &PW substitution parameters to pass information to other Oracle products, SQL*Menu passes / as your username, and nothing as your password, when you invoke those products from a menu application.

Refer to the *ORACLE RDBMS Database Administrator's Guide* for more information on how to employ this feature.

Remote Login

Through SQL*Net, you can specify a remote database when you log in to any SQL*Menu component. When you use the remote login feature, the remote database is transparent to you—SQL*Menu behaves the same whether it is accessing a node across the country or accessing the database on your local machine.

To specify a remote database from the command line, append a driver name and a node name to your password using the following syntax:

```
:node name
```

where:

driver_name	Indicates the driver that you want to use to access the remote database. The default setting is system dependent.
node_name	Indicates the remote network node that contains the target database. You will log into the default database on this node.

This accesses the named database. You will see the applications that you are authorized to use.

To run an application, SQL*Menu requires a local copy of the application. Having a local copy reduces network traffic by making the applications local, while the data is remote.

For example, you might specify the following command to run the sample Orders Application on the default database of the LONDON node:

```
RUNMENU ORDERS SCOTT/TIGER@D:LONDON
```

If you want to specify a remote database from a SQL*Menu login screen, append the driver name and node name to your username (for example, `SCOTT@D:LONDON`).

For more information about SQL*Net, see the SQL*Net documentation for the appropriate communication protocol.

Note: To perform a remote login, you must have an account on the remote computer node where the target database resides.

SQL*Menu System Components

SQL*Menu Version 5.0 has four system components, which can be invoked from the operating system command line by the following commands:

Component	*Command*
SQL*Menu (Design)	SQLMENU
SQL*Menu (Run Menu)	RUNMENU
SQL*Menu (Generate)	GENMENU
SQL*Menu (Document)	DOCMENU

Within SQL*Menu (Design), you can also invoke the components SQL*Menu (Run Menu), SQL*Menu (Generate), and SQL*Menu (Document) by selecting menu items in the design interface.

For information about logging into SQL*Menu (Design) and selecting the SQL*Menu system components through the design interface, see Chapter 5, "Design Operations."

You can display online syntax information for the SQL*Menu commands by entering the command name with the `-?` switch at the operating system prompt.

The following sections describe the SQL*Menu components and their command syntax.

SQL*Menu (Design)

SQL*Menu (Design) provides an interface for creating and modifying menu applications, and for maintaining SQL*Menu files and security tables. It also invokes SQL*Menu (Run Menu), SQL*Menu (Generate), and SQL*Menu (Document).

Syntax

The SQLMENU command has the following syntax:

```
SQLMENU [username/password] [-c [filename:]mapping]
    [-r filename] [-w filename] [-e filename]
```

Arguments

username	Specifies a valid ORACLE username (followed by a slash and the password). If you omit this option, you must enter your username and password in the login screen.
password	Specifies the password to the user's account.

Switches

`-c [filename: ] mapping`	Identifies the Oracle *Terminal resource file *(filename)* and the mapping *(mapping)* to use instead of the default specified at installation. For information about terminal definitions, see the *Oracle *Terminal User's Guide.*
`-r filename`	Reads input from a file instead of the terminal.
`-w filename`	Writes output to a file instead of the terminal.
`-e filename`	Echoes input keystrokes to a file.

To display the command syntax for SQL*Menu (Design), enter the command `SQLMENU-?` at the operating system prompt.

SQL*Menu (Run Menu)

SQL*Menu (Run Menu) runs a menu application by displaying menus and executing the commands behind menu items.

You can invoke SQL*Menu (Run Menu) either from the SQL*Menu (Design) interface or from the command line.

- In the SQL*Menu (Design) interface, select the Execute item in the Action menu.
- On your operating system's command line, use the RUNMENU command.

Syntax

The RUNMENU command has the following syntax:

```
RUNMENU [applic_name] [username/password] [-c [filename:]mapping]
[-m {p |b |f}] [-q] [-s] [-z] [-r file] [-w file] [-e file]
```

Arguments

applic_name	Identifies the name of the application to run. If you omit this option, SQL*Menu displays the Application menu.
username	Specifies a valid ORACLE username (followed by a slash and the password). If you omit this option, you must enter your username and password in the login screen.
password	Specifies the password to the user's account.

Switches

`-c` *[filename:] mapping*	Identifies the Oracle *Terminal resource file *(filename)* and the mapping *(mapping)* to use instead of the default specified at installation. For information about terminal definitions, see the *Oracle*Terminal User's Guide*.
`-m` *style*	Selects a menu display style: • `p` for pull-down (the default style) • `b` for bar style • `f` for full-screen display For descriptions of the different menu display styles, see Chapter 3, "Menu Styles and Navigation."
`-q`	Activates "quiet" mode, suppressing terminal sounds.
`-s`	Suppresses the SQL*Menu banner display.
`-z`	Security: SQL*Menu does not store the password internally. (The substitution parameter &PW has a value of NULL.)
`-r` *file*	Reads input from a file instead of the terminal.
`-w` *file*	Writes output to a file instead of the terminal.
`-e` *file*	Echoes input keystrokes to a file.

If you specify a username and password, you will not see the login screen. If you specify an application name, SQL*Menu (Run Menu) runs that menu application; but you can still access other applications by pressing [Application Menu].

To display the command syntax for SQL*Menu (Run Menu), enter the command `RUNMENU-?` at the operating system prompt.

SQL*Menu (Generate) SQL*Menu (Generate) creates the executable menu library files defined by an application. You can also use SQL*Menu (Generate) to grant access to a new user or revoke user access.

You can invoke the SQL*Menu (Generate) component either from the SQL*Menu (Design) interface or from the operating system command line.

- In the SQL*Menu (Design) interface, select the following items:
  - Generate in the Action submenu generates a menu application.
  - Unload in the Action submenu exports a menu application.
  - Grant Access in the Admin submenu of the Action submenu grants or revokes user access.
- On the operating system's command line, enter the GENMENU command.

Syntax The GENMENU command has the following syntax:

```
GENMENU [applic_name] username/password
[[-e] |[{-g |-ge |-gd |-ga |-r} username]] [-s]
```

Arguments

applic_name	Identifies the name of the application to generate or export. This argument is required if you use the -e stitch.
username	Specifies a valid ORACLE username (followed by a slash and the password).
password	Specifies the password to the user's account.

Switches

-e	Export a application Generates an export file *filename*.SQL, where *filename* is the application's file name. **Note:** To specify the application, you must include the *[applic_name]* argument before the -e switch.
-g *username* or -ge *username*	Grant execute privileges: Grants a new user SELECT access to the SQL*Menu base tables. This switch is equivalent to the Execute privilege level in the Grant Access to User dialog box.
-gd *username*	Grant design privileges: Grants a new user SELECT, UPDATE, INSERT, and DELETE access to the SQL*Menu base tables. This switch is equivalent to the Design privilege level in the Grant Access to User dialog box.

`-ga` *username*	Grant administrate privileges: Grants a new user SELECT, UPDATE, INSERT, DELETE, and GRANT access to the SQL*Menu base tables. This switch is equivalent to the `Administrate` privilege level in the Grant Access to User dialog box.
`-r` *username*	Revoke privileges for *username*.
`-s`	Suppress the SQL*Menu banner display.

For example, the following call generates a library file named *applic_name*. DMM for the menu application *applic_name:*

```
GENMENU applic_name system/menudba
```

The following call grants a new user SELECT access to the SQL*Menu tables, but no UPDATE, INSERT, or DELETE access:

```
GENMENU system/menudba -G newuser
```

The following call grants a new user SELECT, UPDATE, INSERT, and DELETE access:

```
GENMENU system/menudba -GD newuser
```

Note: The switches `-e, -r, -g, and -gx` are mutually exclusive.

To display the command syntax for SQL*Menu (Generate), enter the command `GENMENU-?` at the operating system prompt.

SQL*Menu (Document)

SQL*Menu (Document) generates documentation for a menu application. It can generate either a detailed overview or a summary overview, which omits the item help information. It can also generate documentation for references to a specific object in the application.

You can invoke the SQL*Menu (Document) component from either the SQL*Menu (Design) interface or the operating system's command line.

- In the SQL*Menu (Design) interface, select the Print Doc item in the Action submenu and select the Detail, Summary, or Object Ref item.
- At the operating system prompt, use the DOCMENU command.

Syntax

The DOCMENU command has the following syntax:

```
DOCMENU [applic_name] [username/password] [-b | -f | -o] [-s]
```

Arguments

applic name	Identifies the name of the application to document.
username	Specifies a valid ORACLE username (followed by a slash and the password).
password	Specifies the password to the user's account.

Switches

`-b`	Brief application information, omitting help text. This switch is equivalent to the Summary item in the Print Documentation submenu of the design interface. Brief documentation is the default.
`-f`	Full application information, including all help text. This switch is equivalent to the Detail item in the Print Documentation submenu of the design interface.
`-o`	Object reference. This switch is equivalent to the Object Ref item in the Print Documentation submenu of the design interface.
`-s`	Suppress the SQL*Menu banner display.

If you omit the application name or user identification, SQL*Menu (Document) prompts you for the information.

SQL*Menu (Document) generates a document file named *filename.* DOC, where *filename* is the application's filename (the default) or the filename specified in the File field of the Print Documentation dialog box in the design interface.

To display the command syntax for SQL*Menu (Document), enter the command `DOCMENU -?` at the operating system prompt.

SQL*Menu Constraints

This section discusses limitations *on* various objects in SQL*Menu. Other limits may be imposed by your operating system. For information about system-specific limits, see the *ORACLE Installation and User's Guide* for your system.

Substitution Parameters You can create up to 58 substitution parameters within an application. This number does not include standard (predefine) substitution parameters.

Menu Items There is no fixed limit to the number of items that can be included on each menu. Depending upon the equipment used, the number of items that can appear on a single screen may vary. You can scroll the menu to see additional items.

Menus You can have up to 270 menus in an application.

Levels of Menu Nesting Menus can be nested to a depth of eight to ten levels, depending on your operating system. (See the *ORACLE Installation and User's Guide* for the exact number for your operating system.) It is good practice to go no further than four levels.

Applications SQL*Menu imposes no limit on the number of applications that you can build.

Terminal Definitions

The SQL*Menu DBA may need to define new terminals and maintain current terminal definitions. The DBA also provides SQL*Menu users with lists of the current function key mappings for their terminals.

SQL*Menu Version 5.0 uses Oracle*TerminaI for terminal definition. The default terminal definition for SQL*Menu maps escape sequences into logical functions for a particular terminal.

Oracle*Terminal lets the database administrator redefine function key mappings for an existing terminal or define new mappings for new terminals. For information about terminal definitions, see the *Oracle*Terminal User's Guide.*

Views of the SQL*Menu Base Tables

The SQL*Menu *base tables* store the data for all applications currently installed in the database. SQL*Menu uses the data in the base tables to create application library files, which are accessed whenever a user runs an application, and to maintain application security. *Views* display the information contained in the base tables.

Note: All changes to the data in the base tables should be made through the SQL*Menu (Design) interface.

This section describes the following views of the SQL*Menu base tables, listing their columns and data specifications:

• MENU_V_APPL	Application definitions
• MENU_V_INFO	Menu definitions
• MENU_V_OPTION	Item definitions
• MENU_V_PARAM	Substitution parameter definitions
• MENU_V_PARM_XREF	Menu parameter cross-references
• MENU_V_OBJ_TEXT	Object text information
• MENU_V_GROUP	Role definitions
• MENU_V_GRP_PRIV	Item access privileges
• MENU_V_PRIV	Privilege cross-references
• MENU_V_USER	User enrollment information
• MENU_V_REF	Reference object information
• MENU_V_PROCEDURE	Procedure definitions

MENU_V_APPL

This view contains information about an application. The columns correspond to the fields in the Application Definition form.

APPLICATION_NAME	CHAR(30) NOT NULL
SHORT_NAME	CHAR(15) NOT NULL
FILE_NAME	CHAR(30) NOT NULL
CREATION_DATE	DATE NOT NULL
CREATOR	CHAR(30) NOT NULL
VERSION_RELEASE_NR	NUMBER NOT NULL
LAST_RELEASE_DATE	DATE
MENU_DIRECTORY	CHAR(50)
IDENTIFICATION	CHAR(40) NOT NULL

MENU_V_INFO

This view contains menu information. The columns correspond to fields in the Menu Definition form or spread table, plus the application name.

MENU_NAME	CHAR(30) NOT NULL
APPLICATION_NAME	CHAR(30) NOT NULL
TITLE	CHAR(40) NOT NULL
SUB_TITLE	CHAR(40) NOT NULL
BOTTOM_TITLE	CHAR(72)
OBJECT_TEXT_ID	NUMBER NOT NULL

OBJECT_TEXT_ID indexes the text in the Purpose scroll region of the Menu Definition form or spread table.

MENU_V_OPTION

This view contains menu item ("option") information. The columns correspond to the elements of the Item Definition form or spread table, plus the application and menu names.

MENU_NAME	CHAR(30) NOT NULL
APPLICATION_NAME	CHAR(30) NOT NULL
OPTION_NUMBER	NUMBER NOT NULL
SHORT_NAME	CHAR(15) NOT NULL
DISPLAYED	CHAR(1) NOT NULL
OPTION_TEXT	CHAR(70) NOT NULL
OBJECT_TEXT_ID	NUMBER NOT NULL
COMMAND_TYPE	NUMBER NOT NULL
COMMAND_LINE	CHAR(240) NOT NULL

OBJECT_TEXT_ID indexes the text in the Help Text scroll region of the Item Definition form or spread table, and SHORT_NAME corresponds to the Short Item Name field.

MENU_V_PARAM This view contains the specifications for every substitution parameter. The columns correspond to the elements of the Parameter Definition form or spread table, plus the application name.

SUBSTITUTION_STRING	CHAR(2) NOT NULL
APPLICATION_NAME	CHAR(30) NOT NULL
PAR_SIZE	NUMBER NOT NULL
PAR_DEF	CHAR(64)
ECHO	CHAR(1) NOT NULL
MUST_FILL	CHAR(1) NOT NULL
RESPONSE_REQUIRED	CHAR(1) NOT NULL
UPPERCASE	CHAR(1) NOT NULL
OBJECT_TEXT_ID	NUMBER
PROMPT	CHAR(40)

PAR_DEF corresponds to the Default field of the Parameter Definition form or spread table, and OBJECT_TEXT_ID indexes the text in the Hint field. OBJECT_TEXT_ID is optional for substitution parameters.

MENU_V_PARM_XREF This view contains cross-reference information about substitution parameters for full-screen menus.

MENU_NAME	CHAR(30) NOT NULL
APPLICATION_NAME	CHAR(30) NOT NULL
SUBSTITUTION_STRING	CHAR(2) NOT NULL

MENU_V_OBJ_TEXT This view contains text information associated with SQL*Menu objects, such as internal comments for developers or runtime hints.

APPLICATION_NAME	CHAR(30) NOT NULL
OBJECT_TEXT_ID	NUMBER NOT NULL
OBJECT_TEXT_ORDER	NUMBER NOT NULL
OBJECT_TEXT	CHAR(78)

OBJECT_TEXT_ID indexes the object associated with the text. For text that can be more than one line long, OBJECT_TEXT_ORDER indicates the line order.

MENU_V_GROUP

This view contains information about roles (GROUP_NAME) and their privileges to use the application options DBG (Debug Mode), OSC (Operating System Command), and BGM (Background Menu). The columns correspond to the elements of the Role Definition form or spread table.

GROUP_NAME	CHAR(30) NOT NULL
DEBUG_ALLOWED	CHAR(1) NOT NULL
OS_COMM_ALLOWED	CHAR(1) NOT NULL
BGM_ALLOWED	CHAR(1) NOT NULL
OBJECT_TEXT_ID	NUMBER NOT NULL

OBJECT_TEXT_ID indexes the text in the Comments scroll region of the Role Definition form or spread table.

MENU_V_GRP_PRIV

This view contains information about roles and their menu item privileges. It lists the roles (GROUP_NAME) that have access to each menu item, identified by an internal identification number (PRIVILEGE_ID). This information corresponds to the entries in the Item Role window for each menu item.

APPLICATION_NAME	CHAR(30) NOT NULL
GROW_NAME	CHAR(30) NOT NULL
PRIVILEGE_ID	NUMBER NOT NULL

MENU_V_PRIV

This view contains cross-reference information about menu item privileges. It lists the internal identification number (PRIVILEGE_ID) for each menu item (OPTION_NUMBER).

PRIVILEGE_ID	NUMBER NOT NULL
PRIVILEGE_TYPE	CHAR(3) NOT NULL
APPLICATION_NAME	CHAR(30) NOT NULL
MENU_NAME	CHAR(30) NOT NULL
OPTION_NUMBER	NUMBER NOT NULL

MENU_V_USER

This view contains information about user enrollment. It lists the members (USER_NAME) of each role (GROUP_NAME). This information corresponds to the entries in the Username window for each role.

GROUP_NAME	CHAR(30) NOT NULL
USER_NAME	CHAR(30) NOT NULL

MENU_V_REF This view contains information about reference objects. The columns correspond to the fields in the Copy/Reference Object dialog box, plus the application name.

APPLICATION_NAME	CHAR(30) NOT NULL
REF_TYPE	NUMBER NOT NULL
REF_APP_NAME	CHAR(30) NOT NULL
REF_MENU_NAME	CHAR(30)
REF_OPT_NAME	CHAR(15)
REF_PARM_NAME	CHAR(2)
REF_PROC_NAME	CHAR(30)
NEW_MENU_NAME	CHAR(30)
NEW_OPT_NAME	CHAR(15)
NEW_PARM_NAME	CHAR(2)
NEW_PROC_NAME	CHAR(30)

REF_OPT_NAME and NEW_OPT_NAME correspond to the Item fields for the source and target objects, respectively.

MENU_V_PROCEDURE This view contains information about PL/SQL procedures. It lists all the procedure names and the applications that use them, and indexes the object text for each procedure (OBJECT_TEXT_ID). The columns correspond to the elements of the Procedure Definition form or spread table, plus the application name.

APPLICATION_NAME	CHAR(30) NOT NULL
PROCEDURE_NAME	CHAR(30) NOT NULL
OBJECT_TEXT_ID	NUMBER NOT NULL

PART

V

APPENDIXES

APPENDIX

A UPGRADING SQL*MENU

This appendix explains how to upgrade menu applications created in SQL*Menu Version 4.1 to SQL*Menu Version 5.0.

Upgrading Menu Applications

SQL*Menu Version 5.0 is upwardly compatible with SQL*Menu Version 4.1. When you install SQL*Menu Version 5.0, you can select an option to upgrade existing menu applications.

If you select the upgrade option, SQL*Menu automatically converts all Version 4.1 applications to Version 5.0 applications.

Upgrading One Application

The SQL script CONVMENU lets you convert SQL*Menu applications after you have installed SQL*Menu Version 5.0. This script upgrades application information in the Version 4.1 base tables and inserts the upgraded information into Version 5.0 base tables on the same database.

You can specify the name of the application you want to convert, or you can convert all applications in the Version 4.1 base tables by specifying a percent sign (%) as the application name.

If you want to upgrade a Version 4,1 application that is not on the database where you installed SQL*Menu Version 5.0, first move the application to the database and add it to the Version 4.1 base tables (as described in Chapter 8 of the Version 4.1 *SQL*Menu User's Guide).* Then you can upgrade it to Version 5.0 by using the SQL script:

```
SQLPLUS username/password @CONVMENU application_name
```

Security System

When SQL*Menu converts Version 4.1 applications, it sets up a new security system based on the work-classes defined in the old applications:

- For each SQL*Menu Version 4.1 work-class numbered *xx*, the upgrade creates a role named *appname_xx* where *appname* is the name of the menu application.
- Members are automatically assigned to roles.
- Each converted role has the minimum privileges that the members of the old work-class were granted.

In SQL*Menu Version 4.1, privileges to use menu items are granted to a range of work-class numbers, so that all members of a work-class have identical item privileges. Their option privileges, however, can vary from member to member in SQL*Menu Version 4.1.

In SQL*Menu Version 5.0 the new roles have the same item privileges as their corresponding work-classes, but option privileges for BGM, OSC, and DBG might differ.

For example, if a SQL*Menu Version 4.1 application named WEATHER had three users in work-class 20, then upgrading WEATHER to SQL*Menu Version 5.0 would create a role named WEATHER 20 with only those option privileges that were granted to all three members of the old work-class.

In the following table, since USER3 cannot use the OSC option in SQL*Menu Version 4.1, then none of the WEATHER_20 users have OSC access in SQL*Menu Version 5.0.

Version 4.1

UserID	*Work Class*	*BGM*	*OSC*	*DBG*
USER1	20	Y	Y	N
USER2	20	Y	Y	N
USER3	20	Y	N	Y

Version 5.0

Role	*BGM*	*OSC*	*DBG*
WEATHER_20	Y	N	N

You can modify the membership and privileges of a converted role in the usual way, if necessary.

For more information about SQL*Menu security, see "Security Operations" in Chapter 5.

National Language support

The message translation facility included in SQL*Menu Version 4.1 has been removed in Version 5.0. SQL*Menu Version 5.0 supports the multi-lingual message facility provided by National Language Support (NLS).

If you want to customize the message translations, you can use the National Language Support WorkBench (NLS*WorkBench).

APPENDIX

B

ERROR MESSAGES

This appendix lists the error messages and warnings you might receive as a result of running SQL*Menu. Each message is followed by a description of the cause of the error and what action you can take to recover from it.

This appendix covers the following topics:

- SQL*Menu (Run Menu) Error Messages
- SQL*Menu (Document) Error Messages
- SQL*Menu (Generate) Error Messages
- SQL*Menu (Design) Validation Error Messages
- PL/SQL Compilation Error Messages

The following classes of error messages can appear while you use SQL*Menu:

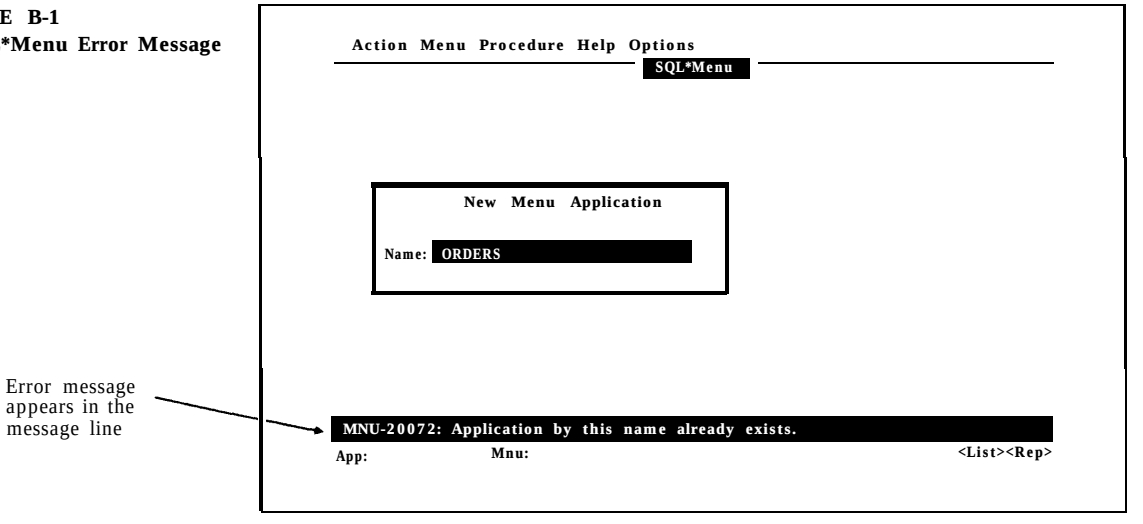
MNU-102 *xx*	SQL*Menu (Run Menu)
MNU-l04 *xx*	SQL*Menu (Document)
MNU-106 *xx*	SQL*Menu (Generate)
MNU-20 *xxx*	SQL*Menu (Design)
MNU-210 *xx*	PL/SQL compilation

For descriptions of the SQL*Menu components, see "SQL*Menu System Components" in Chapter 12.

You might also see error messages for SQL*Forms (FRM- *xxxxx),* if you are using it with SQL*Menu, ORACLE (ORA- *xxxxx),* or PL/SQL (ORA-06503:PL/SQL: error *xxx).* For an explanation of the SQL*Forms error messages, see *SQL*Forms Designer's Reference.* For explanations of the ORACLE and PL/SQL error messages, see *ORACLE Error Messages and Codes Manual.*

Error messages appear in the message line, as shown in Figure B-1.

FIGURE B-1
A SQL*Menu Error Message



SQL*Menu (Run Menu) Error Messages

Error messages for SQL*Menu (Run Menu) operations are numbered MNU-102 *xx*.

MNU-10200: Illegal function in this context.

Cause: You pressed a key which is not allowed in this context.

Action: Press [Show Keys] to see a list of valid function keys.

MNU-10201: No parameters needed.

Cause: You pressed [Enter Application Parameters] or [Enter Menu Parameters] but none are required in this context.

Action: No action required.

MNU-10202: Menus are nested too deep.

Cause: You tried to select an item that would nest menus more than 10 deep.

Action: Press [Main Menu] to return to the main menu and then navigate to the menu you wanted to go to again.

MNU-10203: Selected item is not in this menu.

Cause: In a full-screen menu, you entered a choice number greater than the maximum number of menu items.

Action: Choose an item that is on this menu.

MNU-10204: No command defined for the selected background item.

Cause: You pressed [Background Menu *n]* where *n* was greater than the maximum choice on the background menu.

Action: No action required. Press [Show BGM] to see the valid background menu items.

MNU-10205: Menu *name* not found.

Cause: In the choice field of a full-screen menu, you entered a menu name that does not exist in this application or is not found in the library.

Action: No action is required if the menu does not exist in the application. If it does, regenerate the library.

MNU-10206: Dynamic memory exhausted.

Cause: SQL*Menu has run out of memory.

Action: Contact your DBA.

MNU-10207: No background menu present.

Cause: You pressed [Show BGM], but no background menu exists.

Action: No action required.

MNU-10208: Parameter *name* not found.

Cause: A menu item referenced an undefined parameter.

Action: Contact your DBA.

MNU-10209: No 'next menu' from background in this context.

Cause: The application attempted to navigate to a named menu from the background menu.

Action: No action required.

MNU-10210: Response required.

Cause: You did not enter a required parameter, or you left the choice field blank in a full-screen menu.

Action: Make an entry.

MNU-10211: Field must be filled completely.

Cause: You partially entered a parameter that must be entered completely.

Action: Enter enough data to completely fill the field.

MNU-10212: Login failed for this username and password.

Cause: You specified an illegal username and password.

Action: Check your username and password and try again.

MNU-10213: Login procedure terminated.

Cause: You failed to log in to ORACLE three times in a row.

Action: No action required. If you area valid user, check your username and password.

MNU-10214: No authorization to run any application.

Cause: You are not a valid user of any application in SQL*Menu.

Action: No action required. If you think you should be a valid user, ask your DBA to grant you access to the application you wish to run.

MNU-10215: No help available.

Cause: You pressed [Help] but none is available for this item.

Action: No action required.

MNU-10216: Failed to spawn a command to the operating system.

Cause: The operating system could not spawn a sub-process.

Action: Refer to the error message that the operating system issued.

MNU-10217: No authorization for any item in selected menu.

Cause: You tried to move to a menu that has no items that you can access.

Action: Check the menu name you entered and try again.

MNU-10218: Error for menu *name*.

Cause: SQL*Menu could not read the library information for this menu, or an invalid menu name was specified.

Action: Regenerate the library or correct the menu name.

MNU-10219: Item number is invalid.

Cause: In a full-screen menu, you entered an invalid number in the choice field.

Action: Check the item number and re-enter it.

MNU-10220: No detailed help available for this item.

Cause: You pressed [Help] but none is available for this menu item.

Action: No action required.

MNU-10221: Could not open file *name*.

Cause: Either file privileges are set incorrectly, or the library you tried to open is bad.

Action: Regenerate the application library and try again.

MNU-10222: Library *name* too old, regenerate it.

Cause: You are using a newer version of SQL*Menu than the version that created this library.

Action: Regenerate the application library and re-execute the command.

MNU-10223: Application parameter module does not exist.

Cause: The parameter information could not be located in the library. This may be due to a library file that is bad or one that contains a different application.

Action: Generate the application library and try again. If this is unsuccessful, contact your DBA.

MNU-10224: Application bind variable module does not exist.

Cause: The bind variable information could not be located in the library. This may be due to a bad library file.

Action: Generate the application library and try again. If this is unsuccessful, contact your DBA.

MNU-10225: Could not read parameter data.

Cause: The application library is bad.

Action: Generate the application library and try again. If this is unsuccessful, contact your DBA.

MNU-10226: Could not read bind variable data.

Cause: The application library is bad.

Action: Generate the application library and try again. If this is unsuccessful, contact your DBA.

MNU-10227: Too many menu parameters.

Cause: The application contains more menu parameters than can be used on your operating system.

Action: Revise and regenerate the application or contact your DBA.

MNU-10228: Could not read help text.

Cause: The application library is bad.

Action: Generate the application library and try again. If this is unsuccessful, contact your DBA.

MNU-10229: Could not close file *name.*

Cause: Operating system error or internal error.

Action: Contact your DBA.

MNU-10230: Application procedure module does not exist.

Cause: The procedure information could not be located in the library. This may be due to a bad library file.

Action: Generate the application library and try again. If this is unsuccessful, contact your DBA.

MNU-10231: Could not read procedure data.

Cause: The application library is bad.

Action: Generate the application library and try again. If this is unsuccessful, contact your DBA.

MNU-10233: Navigational procedures/macros not valid in current menu style.

Cause: You tried to use the full-screen, navigational packaged procedures or macros in the pull-down or bar menu display style.

Action: Notify your DBA. You can navigate by following the menu tree, or rerun the application in the full-screen display style.

MNU-10234: Semicolon missing in macro statement.

Cause: The command line specified for this item has a syntax error.

Action: Notify your DBA.

MNU-10235: Macro *name* not found.

Cause: The menu designer specified a macro to be executed which is not defined.

Action: Notify your DBA.

MNU-10236: No procedure/macro specified.

Cause: The menu designer has specified a blank command.

Action: Notify your DBA.

MNU-10237: Argument(s) not allowed for this procedure/macro.

Cause: The menu designer specified an argument to a command that does not take arguments.

Action: Notify your DBA.

MNU-10238: Error executing *name*. Check arguments.

Cause: The menu designer specified an invalid argument or specified too many arguments for a packaged procedure or macro.

Action: Notify your DBA.

MNU-10239: Can't read form by that name.

Cause: SQL*Menu tried to read a form that does not exist in the current directory.

Action: Notify your DBA.

MNU-10240: Form name not specified.

Cause: A RUNFORM command did not give the name of a form to execute.

Action: Notify your DBA.

MNU-10241: Illegal operation when SQL*Forms is active. *name*

Cause: The menu designer specified a packaged procedure or macro that cannot be used when SQL*Forms calls SQL*Menu.

Action: Notify your DBA.

MNU-10242: Cannot call linked-in SQL*Forms from SQL*Forms.

Cause: The menu designer specified a call to linked-in SQL*Forms from within SQL*Forms.

Action: Notify your DBA.

MNU-10243: Error occured during invocation of SQL*Forms.

Cause: A call to SQL*Forms (Run Form) failed.

Action: Notify your DBA.

MNU-10244: Application *name* does not exist.

Cause: The application you specified does not exist in the database, or you do not have access privilege to it.

Action: Check the application name and try again, or contact your DBA.

MNU-10245: Already on first field.

Cause: You pressed [Previous Field] from the first field in the parameter form.

Action: No action required. You cannot go to a field prior to the first field in a parameter form.

MNU-10246: Error executing packaged procedure – inactive form.

Cause: The menu designer specified a packaged procedure that cannot be executed in the current context.

Action: Notify your DBA.

MNU-10247: No active items in root menu of application.

Cause: You tried to open an application but its root menu has no items that you can access. The root menu is either the application's main menu or another menu specified when SQL*Forms called SQL*Menu.

Action: Notify your DBA.

MNU-10248: No direct menu selection allowed when using a root menu.

Cause: The root menu is not the application's main menu, because SQL*Forms specified another root menu when calling SQL*Menu.

Action: No action required. You can only use direct menu selection when the application's main menu is the root menu.

MNU-10249: No authorization to run application name.

Cause: You are not a valid user of the application you tried to run.

Action: No action required. If you think you should be a valid user, ask your DBA to grant you access to the application.

MNU-10250: Error initializing RUNMENU application.

Cause: An internal error occurred.

Action: Notify your DBA.

MNU-10251: Unsupported command type 4 switch used (-e, -i, -r, -w).

Cause: You specified a switch that is not supported when calling SQL*Forms linked in.

Action: Notify your DBA.

MNU-10252: Unknown command type 4 switch used.

Cause: You specified a switch that SQL*Forms does not understand.

Action: Notify your DBA.

MNU-10254: Could not open file for screen shot.

Cause: The operating system had trouble opening a file for some reason (e.g., permission problems, lack of disk space).

Action: Resolve the operating system condition that caused the error.

MNU-10255: Error occurred during printing of screen shot.

Cause: The operating system had trouble with a file.

Action: Resolve the operating system condition that caused the error.

MNU-10256: User is not authorized to run SQL*Menu.

Cause: You are not enrolled in SQL*Menu. You do not have SELECT permission on the SQL*Menu base tables.

Action: Notify your DBA.

MNU-10257: User is not authorized to select specified option.

Cause: You tried to select a menu item to which you do not have access.

Action: Choose another item or notify your DBA.

MNU-10258: Specified menu is already active.

Cause: You tried to navigate to the current menu.

Action: No action required.

SQL*Menu (Document) Error Messages

Error messages for SQL*Menu (Document) are numbered MNU-104 *xx*. These messages concern printing SQL*Menu (Design) documentation, or storing documentation in a file.

MNU-10400: Login procedure terminated.

Cause: You failed to log in to ORACLE three times in a row.

Action: No action required. If you area valid user, check your username and password.

MNU-10401: Can't open file *name* for output.

Cause: The file that would contain documentation could not be opened by SQL*Menu. This maybe due to lack of disk space or system privileges.

Action: Notify your DBA.

MNU-10402: No such application present.

Cause: You attempted to create documentation for an application that does not exist, or you have no access to the application you specified. Or you logged in to SQL*Menu and specified an application that does not exist, or you are not authorized to use it.

Action: Check your application name and try again.

MNU-10409: Cannot close file.

Cause: The documentation file could not be closed. This is an operating system error.

Action: Notify your DBA.

MNU-10410: Wrong descriptor area selected.

Cause: Internal error.

Action: Contact Oracle Customer Support.

MNU-10411: Unable to allocate dynamic memory for descriptor area.

Cause: SQL*Menu ran out of local memory.

Action: Contact your DBA.

MNU-10412: Non-supported ORACLE datatype selected.

Cause: Internal error.

Action: Contact Oracle Customer Support.

SQL*Menu (Generate) Error Messages

Error messages for SQL*Menu (Generate) are numbered MNU-106 *xx*. These messages concern library generation and base table management, including security.

MNU-10601: Options -e, -m, -g, -p, and -r are mutually exclusive.

Cause: You specified more than one of the command line switches -e, -m, -g, -p, and -r. SQL*Menu (Generate) can only execute one of these options at a time.

Action: Re-enter the command using only one of these switches.

MNU-10602: No application name specified.

Cause: You tried to use the -e switch to export an application, but did not specify the application name.

Action: Reenter the command specifying an application name.

MNU-10603: No username to grant/revoke.

Cause: You tried to grant SQL*Menu access to a user, or revoke access, but did not specify the user.

Action: Re-execute the command specifying the username.

MNU-10604: User has no privileges for this application.

Cause: You tried to access an application using a username that is not a member of any role in the application.

Action: Make sure the application name and username are correct and try again. If you are unsuccessful, contact your DBA.

MNU-10605: Application *name* unknown.

Cause: You tried to access an application that does not exist in the database.

Action: Make sure the application name is correct and try again.

MNU-10606: Could not open file *name.*

Cause:. Either file privileges are set incorrectly, or the library you were attempting to open is bad.

Action: Regenerate the application library and try again.

MNU-10607: Library *name* version too old, regenerate it.

Cause: The library you accessed was created by an old version of SQL*Menu.

Action: Regenerate the application library and re-execute the command.

MNU-10608: Could not generate listing of library *name.*

Cause: You tried to use the `-p` switch with the GENMENU command, but either the operating system could not create a listing file because you do not have the privilege, or there was no disk space available, or an internal error occurred.

Action: Make sure you have privilege to create files, and that you have disk space left. If both of these conditions are true, contact Oracle Customer Support.

MNU-10609: Could not close library *name.*

Cause: Operating system error, or internal error.

Action: Contact your DBA.

MNU-10610: ORACLE error *number* occurred, can't continue.

Cause: Internal error.

Action: Contact Oracle Customer Support.

MNU-10611: Could not insert module header.

Cause: The library you tried to insert into is too small. You used the `-m` switch to specify a library size that is too small.

Action: Regenerate the application library using a larger number of modules. If you omit the `-m` switch, SQL*Menu automatically uses the correct number of modules.

MNU-10612: Could not insert menu text.

Cause: You have an operating system error (e.g., lack of disk space).

Action: Resolve the operating system condition that caused the error.

MNU-10613: Could not insert menu parameters.

Cause: You have an operating system error (e.g., lack of disk space).

Action: Resolve the operating system condition that caused the error.

MNU-10614: Could not insert item *number*.

Cause: You have an operating system error (e.g., lack of disk space).

Action: Resolve the operating system condition that caused the error.

MNU-10615: Could not insert procedure for item *number*.

Cause: You have an operating system error (e.g., lack of disk space).

Action Resolve the operating system condition that caused the error.

MNU-10616: Warning: No items defined for menu *name*.

Cause: You are generating an application that has a menu which contains no items.

Action: No action required.

MNU-10617: No more free module headers available.

Cause: You used the `-m` switch to specify a library size that is too small.

Action: Regenerate the application library using a larger number of modules. If you omit the `-m` switch, SQL*Menu automatically uses the correct number of modules.

MNU-10618: Could not insert item help text.

Cause: You have an operating system error (e.g., lack of disk space).

Action: Resolve the operating system condition that caused the error.

MNU-10619: Could not insert application parameter text.

Cause: You ahve an operating system error (e.g., lack of disk space).

Action: Resove the oerating system condition that caused the error.

MNU-10620: Could not insert item overview text.

Cause: You have an operating system error (e.g., lack of disk space).

Action: Resolve the operating system condition that caused the error.

MNU-10621: Could not insert procedure text.

Cause: You have an operating system error (e.g., lack of disk space).

Action: Resolve the operating system condition that caused the error.

MNU-10622: Could not insert bind variable text.

Cause: You have an operating system error (e.g., lack of disk space).

Action: Resolve the operating system condition that caused the error.

MNU-10623: Dynamic memory exhausted.

Cause: SQL*Menu has run out of memory.

Action: Contact your DBA.

MNU-10624: Could not export table *name.*

Cause: The table does not exist, or you do not have privilege to use that table.

Action: Contact Oracle Customer Support.

MNU-10625: Non-supported ORACLE datatype selected.

Cause: Internal error.

Action: Contact Oracle Customer Support.

MNU-10626: Could not generate SQL INSERT statement.

Cause: Internal error.

Action: Contact Oracle Customer Support.

MNU-10627: Circular menus detected. Cannot generate application.

Cause: The menu tree for the application you tried to generate contains loops, in which an item calls a menu above it in the hierarchy.

Action: Restructure your application as a simple menu tree and try again.

MNU-10628: Item references an unidentified menu.

Cause: An item on one of the menus in the application you tried to generate calls a menu that hasn't been defined yet.

Action: Check menu references, and try again.

MNU-10629: Generate errors written to file *name*.

Cause: You had some errors while trying to generate the current menu application.

Action: Check the file to see your errors.

SQL*Menu (Design) Validation Error Messages

Error messages for SQL*Menu (Design) operations are numbered MNU-20 *xxx*. These messages concern validation of application data.

MNU-20001: Application date must be entered.

Cause: You tried to leave the Application Definition form without specifying a creation date for the application.

Action: In the Creation Date field, enter the creation date as DD-MON-YY, where DD is a number for the day, MON is the first three letters of the name of the month, and YY is a number for the year.

MNU-20002: Application creator must be entered.

Cause: You tried to leave the Application Definition form without specifying the creator.

Action: In the Creator field, enter the username of the person creating or modifying the application.

MNU-20003: Version number is invalid.

Cause: You specified a version number with the wrong syntax in the Application Definition form.

Action: In the Version Number field, enter the version as an integer rather than a decimal number.

MNU-20004: Application identification must be entered.

Cause: You tried to leave the Application Definition form without specifying any identifying text.

Action: In the Identification field, enter a description of up to 40 characters for the application.

MNU-20005: Valid application name must be entered.

Cause: You tried to open an application without specifying the application name, or you tried to leave the Application Definition form without entering a name for the application.

Action: Enter a valid application name and try again.

MNU-20006: Valid application short name must be entered.

Cause: You tried to leave the Application Definition form without specifying a valid short name for the application.

Action: In the Short Name field, enter a name of up to 15 characters in length to be used on bar and pull-down style menus.

MNU-20007: Application file name must be entered.

Cause: You tried to leave the Application Definition form without specifying a filename for the application's library file.

Action: In the File Name field, enter a file name in which to store the application's library file. (Do not enter the DMM extension.)

MNU-20008: Invalid application date. Please enter as DD-MON-YY.

Cause: You used an invalid syntax to specify the application creation date.

Action: In the Creation Date field, enter the date as DD-MON-YY, where DD is a number for the day, MON is the first three letters of the name of the month, and YY is a number for the year.

MNU-20009: Invalid release date. Please enter as DD-MON-YY.

Cause: You used an invalid syntax to specify the last release date.

Action: In the Last Release Date field, enter the release date as DD-MON-YY, where DD is a number for the day, MON is the first three letters of the name of the month, and YY is a number for the year.

MNU-20011: Valid menu name must be entered.

Cause: You tried to leave the Menu Definition form or spread table without specifying a menu name.

Action: In the Menu Name field, enter a name of up to 30 characters.

MNU-20012: Menu with this name already exists.

Cause: You entered a menu name that is being used by another menu in this application.

Action: Enter a different name for the menu you are creating or modifying.

MNU-20013: Menu does not exist.

Cause: You entered an invalid menu name in the Menu Name window while associating a parameter with a menu in the Parameter Definition form or spread table.

Action: Check the menu name and re-enter it. If necessary, create the menu in the Menu Definition form or spread table.

MNU-20014: Menu title must be entered.

Cause: You tried to leave the Menu Definition form or spread table without specifying a menu title.

Action: In the Title field, enter a title of up to 40 characters.

MNU-20015: Menu subtitle must be entered.

Cause: You tried to leave the Menu Definition form or spread table without specifying a menu subtitle.

Action: In the Subtitle field, enter a subtitle of up to 40 characters.

MNU-20016: Cannot alter a referenced menu.

Cause: You tried to modify a referenced menu.

Action: No action required. If you need to alter the menu, recreate it or copy it by specifying Copy instead of Reference in the Copy/Reference Object dialog box that appears when you press [Copy Object].

MNU-20018: Cannot insert or delete from a referenced object.

Cause: You tried to modify a referenced object.

Action: No action required. If you need to modify the object, copy it by specifying COPY instead of REFERENCE in the Copy/Reference Object dialog box that appears when you press [Copy Object].

MNU-20021: Item number is invalid.

Cause: You specified an invalid item number **in** the Item Definition form or spread table.

Action: Check the item number and re-enter it in the Item field.

MNU-20022: Item command type is invalid.

Cause: You specified an invalid command type in the Item Definition form or spread table. The command type must be a number from 1 to 7, depending on the type of command entered in the Command Line scroll region.

Action: Enter a number from 1 to 7 in the Command Type field.

MNU-20023: Item text must be entered.

Cause: You tried to leave the Item Definition form or spread table without specifying the text for an item.

Action: In the Item Text field, enter a text description of up to 70 characters for the item.

W-2(X)24 Valid short item name must be entered.

Cause: You tried to leave the Item Definition form or spread table without specifying a valid short name for an item.

Action: In the Short Item Name field, enter a name or description of up to 15 characters for the item. The space character is valid within the short item name, but not as the first character.

MNU-20025: Item with this name already exists.

Cause: You entered an item name that is being used by another item in this menu.

Action: Enter a different name for the item you are creating or modifying in the Short Item Name field.

MNU-20026: Valid command line must be entered for this item.

Cause: You tried to leave the Item Definition form or spread table without specifying the command line for an item.

Action: Enter a command for the item in the Command Line scroll region. The Command Type field for the item must also specify the appropriate number.

MNU-20028: Item command line is too long (240 max).

Cause: You entered a command that contains more than 240 characters in the Item Definition form or spread table.

Action: Reenter the command using a shorter syntax, if possible, or split the command into separate components performed by different menu items.

MNU-20031: Valid parameter substitution string must be entered.

Cause: You tried to leave the Parameter Definition form or spread table without specifying the name for a parameter.

Action: In the Parameter field, enter a unique two-character name.

MNU-20032: Parameter with this name already exists.

Cause: You entered a parameter name that is being used by another parameter in this application.

Action: Enter a different name for the parameter you are creating or modifying.

MNU-20033: Parameter size is invalid.

Cause: You specified an invalid parameter size in the Parameter Definition form or spread table.

Action: In the Size field, enter the parameter size as an integer.

MNU-20035: Default value is too large to fit into specified parameter.

Cause: You entered a default parameter value that is larger than the size specified by the designer in the Size field.

Action: Recheck the size of the parameter, and try again.

MNU-20037: Parameter name must be exactly 2 characters.

Cause: You specified an invalid parameter name in the Parameter Definition form or spread table.

Action: In the Parameter field, enter a name that is exactly two characters long.

MNU-20040: Target menu does not exist.

Cause: You tried to copy or reference an object to a menu that does not exist.

Action: No action required.

MNU-20041: Target menu already exists.

Cause: You tried to copy or reference a menu to a menu that already exists.

Action: Give the target menu a different name. Remember to follow Oracle naming conventions.

MNU-20042: Target item already exists.

Cause: You tried to copy or reference an item to an item with the same name.

Action: Give the target item a different name. Remember to follow Oracle naming conventions.

MNU-20043: Target parameter already exists.

Cause: You tried to copy or reference a parameter to a parameter with the same name.

Action: Give the target parameter a different name. Remember to follow Oracle naming conventions.

MNU-20044: Target procedure already exists.

Cause: You tried to copy or reference a procedure to a procedure with the same name.

Action: Give the target procedure a different name. Remember to follow Oracle naming conventions.

MNU-20045: Source object does not exist.

Cause: You tried to copy or reference an object that does not exist.

Action: No action required.

MNU-20046: Object to be copied/referenced must be entered.

Cause: You neglected to enter source object information.

Action: Enter information and try again.

MNU-20051: Valid role name must be entered.

Cause: You tried to leave the Role Definition form or spread table without specifying the name for a role.

Action: In the Role Name field, enter a name of up to 30 characters.

MNU-20052: Role with this name already exists.

Cause: You entered a duplicate role name in the Parameter Definition form or spread table.

Action: Enter a different name for the role you are creating or modifying.

MNU-20055: Valid username must be entered.

Cause: You entered an invalid username into the role.

Action: Check spelling of username.

MNU-20056: Valid privilege level must be entered.

Cause: You entered an invalid privilege level for this user.

Action: You must enter one of the following:

Execute	(User)
Design	(Designer)
Administrate	(SQL*Menu DBA)
Revoke	(revoke all privileges)

Use the list of values if you need help.

MNU-20061: Valid username must be entered.

Cause: You tried to leave the Username window without specifying a username for the current role.

Action: Enter a valid username to be a member of the current role in the Username field.

MNU-20062: User with this name already exists.

Cause: You entered a username that is already a member of the current role.

Action: Check the spelling of the username, or check which role the Username window currently represents.

MNU-20063: Procedure name is a reserved PL/SQL word.

Cause: You used a reserved word when naming your procedure.

Action: Check the reserved words list.

MNU-20064: Procedure name conflicts with a packaged procedure or constant.

Cause: You duplicated a packaged procedure name while naming your procedure.

Action: Check the list of PL/SQL packaged procedures.

MNU-20065: Procedure name is invalid.

Cause: You entered a procedure name that does not follow Oracle naming conventions.

Action: No action required, try again.

MNU-20066: Cannot rename a referenced procedure.

Cause: In the Copy/Reference Object dialog box, you tried to give a target procedure name for a referenced procedure.

Action: Try again without renaming the procedure.

MNU-20071: Invalid menu application name.

Cause: You entered an application name that does not conform to ORACLE naming conventions while trying to open a new application or save the current application.

Action: Enter a valid ORACLE object name and try again.

MNU-20072: Application by this name already exists.

Cause: You tried to create a new application with the same name as an existing one.

Action: Enter a unique application name. To display a list of the existing applications, press [List].

MNU-20073: Application being copied does not exist.

Cause: You tried to copy an application that does not exist in the database.

Action: Enter the name of an existing application and try again. To display a list of the existing applications, press [List].

MNU-20074: Application being renamed does not exist.

Cause: You tried to rename an application that does not exist in the database.

Action: Enter the name of an existing application and try again. To display a list of the existing applications, press [List].

MNU-20076: Application does not exist.

Cause: You tried to access an application that does not exist in the database.

Action: Check the name of the application and try again.

MNU-20078: Menu not found for reference.

Cause: You tried to reference a menu that could not be located.

Action: Check the menu and application names for the menu you want to reference, and try again. If you are unsuccessful, contact your DBA.

MNU-20079: Error occured during invocation of DOCMENU.

Cause: A call to the DOCMENU component failed.

Action: Contact your DBA.

MNU-20080: Error occurred during printing of DOCMENU document.

Cause: An error occurred while SQL*Menu was printing the application documentation.

Action: Check your system printer or contact your DBA.

MNU-20081: Application locked... Continue at your own risk.

Cause: Another designer is working on the application. This condition can occur after you save the security table, which unlocks your application.

Action: Coordinate with the other designer so that only one of you is working on the application at the same time. You can also try saving the application under a different name if you want to continue working on it immediately.

MNU-20082: Role table locked... Continue at your own risk.

Cause: Another designer is working on the security table. This condition can occur after you save the security table, which unlocks your application.

Action: Coordinate with the other designer so that only one of you is working on role definition at the same time. You can also try saving the application under a different name if you want to continue working on it immediately.

MNU-20084: Application locked... Try again later.

Cause: Another designer is working on the application.

Action Wait until the other designer is finished working on the application, then try again.

MNU-20086: Role table locked... Try again later.

Cause: Another designer is working on the security table.

Action: Wait until the other designer is finished working on the security table, then try again.

MNU-20087: File name for storing screen shot must be specified.

Cause: You pressed the [Print] key, but did not specify a filename in the Print to File dialog box.

Action: Enter a file name in the File field.

MNU-20088: Error opening file for storing screen shot.

Cause: You have an operating system error (e.g., lack of disk space).

Action: Resolve the operating system condition that caused the error.

MNU-20089: Error occurred while printing screen shot.

Cause: You have an operating system error (e.g., lack of disk space).

Action: Resolve the operating system condition that caused the error.

MNU-20091: Invalid PL/SQL procedure name.

Cause: You entered a procedure name that does not conform to ORACLE naming conventions.

Action: Enter a valid procedure name of up to 30 characters and try again.

MNU-20093: You must enter either Copy or Reference.

Cause: You pressed [Copy Object] but did not specify "Copy" or "Reference" in the Copy/Reference Object dialog box.

Action: In the Copy or Reference field, enter "Copy" or "Reference", or press [List] to select a valid entry.

MNU-20094: Out of memory while allocating for menu reference.

Cause: Internal error.

Action: Exit SQL*Menu and restart, or contact your DBA.

MNU-20095: Procedure with this name already exists.

Cause: You entered a duplicate procedure name in the Procedure Definition form or spread table.

Action: Enter a different name for the procedure you are creating or modifying.

MNU-20096: Parameter not found for reference.

Cause: You tied to reference a parameter that does not exist.

Action: Check entry in the Parameter Name field, and try again.

MNU-20097: Procedure not found for reference.

Cause: You tried to reference a procedure that does not exist.

Action: Check entry in the Procedure field and try again.

MNU-20098: Error on compiling procedure specifications.

Cause: Compile error occurred for current procedure.

Action: Resolve the errors that are displayed in the Compilation Errors form.

MNU-20100: Error initializing menu designer application.

Cause: Internal error occurred.

Action: Contact your SQL*Menu DBA.

PL/SQL Compilation Error Messages

Error messages for PL/SQL compilation in SQL*Menu are numbered MNU-210 *xx*. Other PL/SQL errors are detected by ORACLE and labeld `ORA-06503:PL/SQL:error` *xxx*. For an explanation of these ORACLE PL/SQL error messages, see the *ORACLE Error Messages and Codes Manual.*

SQL*Menu can list PL/SQL error messages on your screen or in an error file named *application_file_name* `err`, as described in "PL/SQL Compilation" in Chapter 9.

MNU-21001: Compile error on procedure *name.*

Cause: An error occurred while trying to compile a procedure.

Action: Check the error listings displayed on your screen or stored in the error file, and check the syntax of the procedure text. See the *PL/SQL User's Guide and Reference* to revise the procedure.

MNU-21002: Code given for procedure *name* is not executable.

Cause: The code for this procedure is invalid.

Action: Check the error listings displayed on your screen or stored in the error file, and check the syntax of the procedure text. See the *PL/SQL User's Guide and Reference* to revise the procedure.

MNU-21003: Code given for procedure *name* is not a named unit.

Cause: This procedure contains an anonymous block or invalid code.

Action: Rewrite the anonymous block or invalid code as a procedure with correct syntax. See the *PL/SQL User's Guide and Reference* to revise the procedure.

MNU-21004: Mismatched procedure and unit name for procedure *name.*

Cause: The procedure name does not match the name that defines the procedure in the corresponding procedure text.

Action: Revise the procedure text using the correct procedure name and syntax.

MNU-21005: Compile error on item *name*.

Cause: An error occurred while trying to compile PL/SQL code from the command line of this menu item. (The item *name* is the value of the Short Item Name field in the Item Definition form or spread table.)

Action: Check the error listings displayed on your screen or stored in the error file, and check the code syntax. See the *PL/SQL User's Guide and Reference* to revise the code.

MNU-21006: Code given for item *name* is not executable.

Cause: The code used in the command line of this menu item is invalid. (The item *name* is the value of the Short Item Name field in the Item Definition form or spread table.)

Action: Check the error listings displayed on your screen or stored in the error file, and check the code syntax. See the *PL/SQL User's Guide and Reference* to revise the code.

MNU-21007: Code given for item *name* takes parameters.

Cause: The code used in the command line of this menu item should include parameters, but no parameters are specified. (The item *name* is the value of the Short Item Name field in the Item Definition form or spread table.)

Action: Revise the code, specifying the appropriate parameters.

MNU-21008: Code given for item *name* is a named unit.

Cause: The PL/SQL code in the command line of this menu item should bean anonymous block, but it is a named unit. (The item *name* is the value of the Short Item Name field in the Item Definition form or spread table.)

Action: Use the Procedure Definition form or spread table to make the code a procedure, or revise the code as an anonymous block in the item's command line.

MNU-21009: Compile error on procedure specification for procedure *name*.

Cause: An error occurred during procedure specification.

Action: Check the error listings displayed on your screen or stored in the error file, and see the *PL/SQL User's Guide and Reference* to revise the procedure.

GLOSSARY

access Entry to or communication with a particular object, such as an operating system, the ORACLE RDBMS, or specific files or tables.

application A software program, or set of programs, tailored to achieve a specific result. A *menu application* is a set of interconnected menus created with SQL*Menu (Design) and run by SQL*Menu (Run Menu).

For example, an application to control a company's stock could consist of a set of menus invoking various SQL*Forms screens for data input, and various SQL*ReportWriter programs to produce reports of summary data.

argument An independent variable specified with a command to affect its operation or supply a value.

background menu (BGM) A menu of frequently used commands that can be accessed from any part of a menu application.

block (1) The basic unit of storage for data in the ORACLE RDBMS. (2) A portion of a SQL*Forms form containing a group of fields associated with the columns of a table or view.

branch Part of a menu tree, consisting of interconnected menus derived from one submenu.

button An interface element that can set an option, open a dialog box, or open a list window.

check box An interface element that toggles a setting or value.

command An instruction or request for the system to perform a particular action. An entire command can consist of the command name, parameters, and qualifiers.

command file File containing a predefined sequence of commands to be executed by the operating system.

command line The entire command string, including the command and any parameters or qualifiers it may have.

database administrator A person responsible for the operation and maintenance of the ORACLE RDBMS or of SQL*Menu. The database administrator (DBA) monitors the program's use and customizes it to meet the needs of the local community of users. There may be more than one DBA per site. There may be separate DBAs for ORACLE and for SQL*Menu, or one person maybe responsible for maintaining both programs.

DBA See database administrator.

default A value supplied by the system when a user does not specify a required command parameter or qualifier.

dialog box A pop-up window containing one or more fields for data entry.

directory File cataloging a user's files on a particular storage device.

field An area of data or an area (of a screen or file) reserved for data of a certain type.

file A collection of data treated as a unit, such as a list, document, index, note, or set of procedures. Generally used to refer to data stored on magnetic tapes or disks.

filename The name component of a file specification.

form A set of field definitions, constant text, and other interface elements used for manipulating information in database tables. In SQL*Menu a form displays data from a single record.

function key A keyboard key, or combination of keys, that can perform a particular task. The mapping of keys to functions varies.

interface The hardware and/or software required to connect computers and/or peripherals to allow data flow. Also, a software program (such as SQL*Menu) that lets the user conveniently work with one or more other software programs.

log in To start running a software program using a particular username and password.

macro A simple named procedure in a computer language or program.

main menu The primary menu of an application, giving access to other menus.

menu A list of choices (menu items) available for performing tasks or calling other menus.

menu application A set of interconnected menus created with SQL*Menu (Design) and run by SQL*Menu (Run Menu).

menu item An action choice listed on a menu.

menu tree A set of menus connected hierarchically, with submenus branching from menu items in the main menu and other submenus.

multi-user system A system that can be used by several users concurrently.

nesting Structuring a multi-level system hierarchically, so that each lower-level item is subsidiary to a higher-level item. For example, a menu in a nested structure can have several subsidiary submenus.

null A non-value, meaning that the value is unknown or inapplicable. The value NULL is not equivalent to a zero value, but rather represents a missing value.

object (1) A named group of data in the ORACLE database, such as a menu, substitution parameter, form, or table. (2) A group of data in a menu that you can copy, move, or delete in a single operation.

operating system The system software that controls the computer and its parts, performing basic tasks such as allocating memory and allowing computer components to communicate.

option A variation from the standard mode of running a program. Options can be selected while the program is running or as arguments *(switches)* in the command sting that invokes the program.

parameter Object of a command. A parameter can be a file specification, a symbol value passed to a command procedure, or a word defined by the operating system. In SQL*Menu, *substitution parameters* function as variables for operator data entry and PL/SQL procedures.

password A secondary identification word associated with a username. A user logging in to the system must supply the correct password before the system will permit access. This security measure helps to prevent unauthorized people from working with files.

prompt Word(s) used by the system as cues to assist a user's response. Such messages generally ask the user to respond by typing some information in the following field.

record A unit of information in a form or spread table that corresponds to a row in a database table.

single-user system An operating system that can be accessed by only one user at a time.

spread table A set of field definitions, constant text, and other interface elements used for manipulating information in database tables. Each row of a spread table displays data from a single record.

string Any sequence of words or characters on a single line.

switch A command option that can be specified as an argument in a command string.

syntax The orderly system by which commands, qualifiers, and parameters are arranged together to form valid command strings.

terminal A hardware communication device, with a typewriter-like keyboard that receives and transmits information between users and the system.

username A name by which the system uniquely identifies a particular user; assigned by the system administrator or database administrator. To gain access to the system, a person must specify a username followed by a password.

window A type of interface element that can appear as needed for performing specific tasks. A window overlies part of the screen (including any other windows already displayed), and when it is erased the information it covered appears again. A SQL*Menu window can contain a dialog box, list of values, help information, or other interface elements.

INDEX

!

apl in menu path 3-10
&AD substitution parameter 8-25, 11-6
&PW substitution parameter 8-25, 11-6, 12-5
&SO substitution parameter 8-25
  assigning a value 11-6
&TT substitution parameter 8-25, 11-6
&UN substitution parameter 8-25, 11-6

A

Accelerator key 1-4
[Accept] function key 4-3, 7-3, 11-3, 11-5
  in a dialog box 2-8, 5-9, 5-34 to 5-36, 5-38
  in a form or spread table 5-13 , 6-10, 6-14
  in a window 5-46, 6-6
Access G-1
  application 5-39
  granting 5-16, 5-39 to 5-40, 8-15
  revoking 5-41, 12-7
Action item 5-9
Action submenu
  Admin item 5-40 to 5-42, 5-45, 12-6
  Copy item 5-33
  Delete item 5-36
  Execute item 5-26
  Generate item 5-25, 12-6
  Print doc item 5-27, 12-8
  Quit item 5-4
  Rename item 5-35
  Save item 5-24
  Unload item 5-37, 12-6
&AD substitution parameter 8-25, 11-6
Admin item 5-40
  in Action submenu 12-6
Admin submenu 5-40
  Grant access item 5-40 to 5-41
  Host item 5-36
  Security item 5-42, 5-45
Administrator
  See also Database administrator
  privilege level 5-39

AFORM state 10-3
Alert 6-5
  in the design interface 6-5
  quitting SQL*Menu 5-5
  security table 5-44, 5-46
AMENU state 10-3
Anonymous block 5-22, 9-4
  See also PL/SQL
apl in menu path 3-10
App: status line item 6-5
Application 1-2, G-1
  application menu 2-3, 3-10, 4-6
  backing up 5-37
  copying 5-33
  creating new 5-8, 8-5
  deleting 5-36
  deleting roles 5-44
  deleting user 5-46
  design overview 5-2
  designing 5-3
  directory 5-24
  documenting 5-27, 12-8
  executing 2-2, 5-26
  generating 5-25
  help information 2-10 to 2-11
  item in Menu submenu 5-10
  library file 5-25, 5-36
  menu tree 1-7
  modifying definition 5-10, 8-5
  moving (exporting) 5-37
  name 8-3, 8-8, 12-5 to 12-6, 12-8, B-27
  opening 5-9
  operators and designers ii, 12-6
  options 12-13
  renaming 5-35
  required to select Menu item 5-10, 5-12
  running a report 2-8
  sample 1-5
  saving 5-24
  title 1-6, 8-7 to 8-8
  unlimited number 12-9
  unloading (exporting) 5-37
  upgrading A-2
  version 8-8
Application Definition form 5-10 to 5-11, 12-10
  directory 5-25, 8-7
  error messages B-19 to B-20
  file name 5-25, 5-37, 8-7
  object definition 8-5
  version 8-8
Application designer
  privileges 5-39, 12-6
Application menu
  after [Redefine Usr/Pwd] 4-6
  exiting SQL*Menu (Run Menu) 2-3
  Where option 3-10
[Application Menu] function key 4-3, 11-2, 12-5
  exiting SQL*Menu (Run Menu) 2-3
  selecting an application 3-2
Application operator
  deleting from role 5-46
  privileges 5-39, 12-6
  role 5-45
Application user
  privileges 12-6
APPLICATION_MENU
  packaged procedure 10-5
APPLICATION_PARAMETER
  packaged procedure 10-6
Argument G-1
  macro 11-6
  packaged procedure 10-4
  SQL*Menu components 12-4 to 12-6, 12-8
Arrow indicators in a scroll bar 6-11, 6-16
ASSIGN macro 11-5 to 11-6
  predefine parameter error 11-6
Assigning values to parameters 8-25, 11-6
Authorization
  access privileges 5-39, 12-6
  See also Login screen
  SQL*Menu 1-9
  system username and password 1-9
Auto Restrict List of Values option 5-29, 6-13
Automatic login 12-2

B

Background menu 1-4, 2-12, G-1
  option 2-12, 5-44
  sample application 2-12
Background Menu check box 5-44
[Background Menu *n]* function key 2-12, 4-3, 11-2
Background Menu option 5-44, 12-13
BACKGROUND_MENU *n*
  packaged procedure 10-6
Bar menu 3-5
  application identification 8-7
  invoking 12-5
  item text 8-15
  navigating 3-2
  short item name 8-15
  using 3-3
Base tables 5-39, 12-6, 12-10
[Beginning of Line] function key 4-3, 7-3
BGM
  See Background menu
  See Background Menu option
BGM status line item 1-7, 2-12
  See also Background menu
Bit-mapped environment
  list of values 6-12
  noted differences 6-16
  selecting items 2-4, 6-4
  SQL*Menu functions 4-2, 7-2
Block 9-3, G-1
  anonymous 9-4
  PL/SQL 1-5, 8-25, 9-3
  procedure 9-4
Block-mode environment 2-5, 4-6, 7-9
[Bookmark] function key 6-19, 7-3
Bottom title in full-screen menus 8-11
Branch
  See Menu tree
Button 6-6, G-1
  Cancel button 5-45 to 5-46
  Grant Role Access button 5-15, 8-15
  No button 5-5, 5-45 to 5-46
  radio button 5-30, 6-6
  response to alert 5-45 to 5-46
  Select Menus button 5-19, 8-24
  Select Users button 5-45
  Yes button 5-5, 5-45 to 5-46

C

-c switch 8-18
Cancel button 5-45 to 5-46
[Cancel] function key 3-2, 4-4, 7-4
  in a dialog box 10-3
  in a list of values 6-11
  in the application menu 2-3
CAUTION alert label 6-5
[Change Display Type] function key 5-45, 7-4
  in a form 6-11, 6-16
  in a spread table 6-14, 8-9, 8-13, 8-21, 8-27
Character mode
  environment 6-1, 11-5
Check box 6-8, G-1
  Auto Restrict List of Values 6-13
  Background Menu 5-44
  Debug Mode 5-44
  Display without Privilege 8-14
  Echo 8-23
  Fixed Length 8-23
  in Option Selection form 5-30
  OS Command 5-44
  required 8-24
  Send to Printer 5-28
  Show List of Values 6-13
  Upper Case 8-25
CHRMODE macro 11-5
[Clear Field] function key 4-4, 7-4, 11-3

Command G-1
argument 11-6, 12-4 to 12-6, 12-8, G-1
file G-1
operating system commands 2-14
switch 12-3, G-3
Command line 8-16, 8-25, G-1
menu item 5-16, 10-15
substitution parameter 8-25
username 8-18
Command Line scroll region 8-14,
8-16 to 8-20, 9-4
Command line switch
See Switch
Command syntax
-? switch 12-3
DOCMENU 12-8
GENMENU 12-6
RUNMENU 12-4
SQLMENU 12-4
typography iii
Command types 8-14, 8-16
argument to OSCMD or OSCMD1 macro 11-7
argument to packaged procedure 10-13
error messages B-22 to B-23
execute macro (type 6) 8-19
execute OS command (types 2, 3) 8-17
invoke a submenu (type 1) 8-16
invoke SQL*Forms (type 4) 8-17
invoke SQL*Plus (type 5) 8-18
macro argument 11-7
PL/SQL (type 7) 5-22, 8-20, 9-2
Comments scroll region 5-44, 6-13, 12-13
Compiling PL/SQL 9-9
Compilation Errors form 9-9
Disable PL/SQL Compilation option 5-29
error messages 9-9, B-31
Components
SQL*Menu 12-3
Components of SQL*Menu 1-3
Configuration file
See User preference file
Constraints
Auto Restrict List of Values option 5-29
direct selection in full-screen menus 3-9
displayed items 12-9
macros 11-4
nesting 12-9
number of menus 12-9
objects 12-9
PL/SQL 9-8, 10-2
SQL*Menu 12-9
substitution parameters 12-9
Context 4-3, 6-2
current item 2-4
current menu 5-13, 5-19 to 5-20, 5-22
design interface 6-2
form 6-10
function keys 4-2, 6-2, 7-2
help 6-17
hierarchy 6-2
hint for current item 3-3
object hierarchy 6-2, 8-2
packaged procedure 8-28
required to select Menu item 5-10, 5-12, 5-14
[Show Keys] 4-2, 4-7, 7-2
spread table 6-15
SQL*Menu (Run Menu) help 2-10
SQL*Menu states 10-2, 10-5 to 10-15, 11-4
status line 5-14
unique object names 8-3
Conventions
menu naming 8-3
object naming 8-3
procedure naming 9-5
typographic iii
[Copy] function key 7-4
Copy item 5-33
Copy Menu Application dialog box 5-33
[Copy Object] function key 7-5
Copy/Reference Object dialog box 5-23
error messages B-21, B-29
procedures 9-5
using 5-23

Copy/Reference Object dialog box 5-23, 12-14
  error messages B-21, B-29
  procedures 9-5
Copying an application 5-33
  error messages B-27
Creating a library file 5-25
Creation date 8-6
Creator 8-7
Current context 6-2
  See also Context
Cursor 2-4
  help for current context 2-10
  PL/SQL 9-2
  selecting menu items 2-6
[Cut] function key 7-5

D

Data-entry interface elements 4-5, 6-5
Database administrator ii, 1-2, 1-9, G-1
  privilege level 5-39
  responsibilities 1-9, 5-45, 12-9
  terminal definition 12-9
Date 8-8
DBA
  See Database administrator
DBG
  See Debug Mode option
DBG status line item 1-7
Debug Mode check box 5-44
[Debug Mode] function key 4-4, 5-26, 11-3
Debug Mode option 5-26, 5-44, 12-13
DEBUG_MODE
  packaged procedure 10-6
Default G-2
  creation date 8-6
  creator 8-7
  directory 5-37, 8-7
  display no privilege 8-14
  display style 6-11, 6-16, 10-16
  file name 8-7
  identification 8-7
  item number 8-15
  macro processing 11-5
  menu display style 12-5
  OS command type 10-13
  parameter value 8-23
  short application name 8-8
  summary documentation 12-8
  terminal definition 12-4 to 12-5
Defining an object
  application 5-8, 5-10, 8-5
  menu 5-12, 8-9
  menu item 5-14, 8-12
  parameter 5-17, 8-21
  procedure 5-21, 8-26
DELCHR macro 11-5
[Delete Backwards] function key 4-4, 7-5, 11-3
[Delete Character] function key 4-4, 7-5
Delete item 5-36
[Delete Line] function key 7-5
Delete Menu Application dialog box 5-36
DELETE privilege 5-39, 12-6 to 12-7
[Delete Record] function key 7-5
  in a form or spread table 5-13, 5-44, 6-10, 6-14
  in a window 5-16, 5-46
Deleting an application 5-36

Design interface 6-1
application maintenance operations 5-33
basic operations 5-6
See Interface
running SQL*Menu components 12-3
security operations 5-38
setting options 5-29
Designer ii, 1-2
function keys 7-2
privilege level 5-39, 12-6
Detail item 5-27
equivalent switch 12-8
Dialog box 2-8, 6-8, G-2
Copy Menu Application 5-33
Copy/Reference Object 5-23, 9-5, 12-14
Delete Menu Application 5-36
Enter Parameter Values 2-8, 5-19, 8-25, 10-3
Execute Menu Application 5-26
Export Menu Application 5-37
Generate Menu Application 5-25
Grant Access to User 5-40 to 5-41, 12-6 to 12-7
in the design interface 6-8
New Menu Application 5-8 to 5-9
Open Menu Application 5-9
Print Documentation 5-27
Rename Menu Application 5-35
Save Menu Application 5-24
Search 6-14
Directory G-2
application library file 5-24 to 5-25, 8-7
current 5-37
default 5-37, 8-7
export file 5-37
Disable PL/SQL Compilation option 5-29
DISABLE_ITEM
packaged procedure 10-7
Disabling menu items 10-7
DISP macro 11-5
Display style 3-3, 5-29
bar style 3-5, 8-7, 8-11, 8-15
See also Environment
forms and spread tables 5-42, 6-11, 6-16, 7-11
full-screen 2-4, 3-6, 5-15, 8-11, 8-15, 8-25, 12-5
invoking from SQL*Forms 10-16
invoking full-screen display 12-5
pull-down 3-4, 8-7, 8-11, 8-15
radio button selection 5-26, 5-29
RUNMENU -m switches 3-3, 3-8, 12-5
Display without Privilege check box 8-14
default 8-14
DMM file
generating 5-25
naming conventions 8-3
DOC file
creating 5-28, 12-8
error messages B-12
DOCMENU command
-? switch 12-8
error messages B-12, B-28
SQL*Menu (Document) 12-8
Documenting 5-27
[Down] function key 4-4, 5-14, 5-45, 7-5, 11-3
in a form or spread table 5-20, 6-11, 6-16
in a full-screen menu 2-4, 3-7
in a list of values 6-12
in a pull-down menu 3-2
Dynamic menus
common features 1-6
unavailable items 1-5, 8-14

E

-e switch 8-18
Echo check box 8-23
Elevator
in a scroll bar 6-12, 6-16, 6-18

ENABLE_ITEM
packaged procedure 10-8
Enabling menu items 10-8
[End of Line] function key 4-4, 7-5
[Enter 1 OS Command] function key 4-4, 11-3
using 2-14
[Enter >1 OS Command] function key 4-4, 11-3
using 2-14
[Enter Application Parameters] function key 4-4, 11-2
[Enter Menu Parameters] function key 4-5, 11-3
Enter Parameter Values dialog box
AFORM state 10-3, 10-6
entering values 2-8, 8-25
full-screen menus 5-19, 8-25
IFORM state 10-3
menu state 10-3
MFORM state 10-3, 10-11
parameter processing 5-19, 8-26
Enter Your Choice field 3-6 to 3-8
Environment
bit-mapped 2-4, 4-2, 6-4, 6-12, 7-2
block mode 2-5, 4-6, 7-9
character mode 6-1, 11-5
ERASE
packaged procedure 9-7
Error message
ASSIGN macro 11-6
compilation error file 9-9
DOCMENU B-12, B-28
error types B-2
in message line 2-10, 5-29, 6-4
macro failure 11-4
MNU-102xx B-3
MNU-104xx B-12
MNU-106xx B-14
MNU-20xxx B-19
packaged procedure failure 10-4
PL/SQL compilation 9-9, B-31
PL/SQL Compilation Errors form 9-9
ERROR_CODE packaged function 10-3
ERROR_TEXT packaged function 10-3
ERROR_TYPE packaged function 10-3
Exception handling 9-2, 9-11
Executable statement failure 9-10
packaged procedure failure 9-10
SQL statement failure 9-11
testing statement execution 10-3
Execute 5-26
Generate before Executing Menu option 5-25
item in Action menu 12-4
item in Action submenu 5-26
macro command 8-19
operating system command 8-17
PL/SQL 8-20
Execute Menu Application dialog box 5-26
Exit item 2-3 to 2-4
Exit macro 11-6
EXIT_MENU
packaged procedure 10-9
Exiting SQL*Menu 2-3, 5-4, 8-20
Export
GENMENU -e switch 12-6
Export Menu Application dialog box 5-37
Exporting an application 5-37
SQL file 5-38
See also Unload item

F

Features
SQL*Menu 1-3
Field 6-9, G-2
File field 5-37
Find field 6-12
Hint field 12-12
Name field 5-37
Short Item Name 9-10, 12-11, B-22, B-32
validation 6-10

File G-2
application library 12-10
command file 2-8
compilation error file 9-9
directory path 5-37, 8-7
DMM library file 8-3
DOC file 5-28, 12-8
field in Export dialog box 5-37
filename 5-25, 5-28, 8-7, G-2
input and output 12-5 to 12-6
library 8-7
moving an application 5-38
SQL file 5-38, 12-6
Find field
[Next Field] function key 6-12
in list of values 6-12
[First Line] function key 7-6
Fixed Length check box
parameter value 8-23
Form 6-10, G-2 to G-3
Application Definition 5-10
Compilation Errors 9-9
context 6-10
display style 5-12, 5-29, 5-42, 7-11
Item Definition 5-14
Menu Definition 5-12
Option Selection form 5-25 to 5-26
Procedure Definition 8-26, 8-29
Role Definition 5-42, 5-45
sample order-entry form 3-8
Use Forms as Default option 5-29
FORM_FAILURE packaged function 10-3
FORM_FATAL packaged function 10-3
FORM_SUCCESS packaged function 10-3
Full-screen menu 3-6
application identification 8-7
bottom title 8-11
invoking SQL*Menu (Run Menu) 3-8, 12-5
item text 5-15, 8-15
macro context 11-4
menu parameters 5-18 to 5-19, 8-24 to 8-26
navigating 1-4, 2-4, 3-2, 8-15
restrictions on direct selection 3-9
subtitle 8-11
title 8-12
Where option 3-9, 8-11
Function key 4-2, 7-2, 11-2, G-2
[Accept] 2-8, 4-3, 5-13, 6-6, 6-10, 7-3
[Application Menu] 3-2, 4-3, 12-5
[Background Menu *n*] 2-12, 4-3
[Beginning of Line] 4-3, 7-3
[Bookmark] 6-19, 7-3
[Cancel] 2-3, 3-2, 4-4, 6-11, 7-4, 10-3
[Change Display Type] 5-45, 6-11, 7-4, 8-9
[Clear Field] 4-4, 7-4
context 6-2
[Copy] 7-4
[Copy Object] 7-5, 9-5
corresponding macros 11-2
cursor functions 4-8, 7-12
[Cut] 7-5
[Debug Mode] 4-4, 5-26
[Delete Backwards] 4-4, 7-5
[Delete Character] 4-4, 7-5
[Delete Line] 7-5
[Delete Record] 5-13, 6-10, 7-5
[Down] 3-7, 4-4, 5-14, 5-45, 6-11, 6-16, 7-5
editing functions 4-8, 7-12
[End of Line] 4-4, 7-5
[Enter 1 OS Command] 2-14, 4-4
[Enter >1 OS Command] 2-14, 4-4
[Enter Application Parameters] 4-4
[Enter Menu Parameters] 4-5
[First Line] 7-6
general functions 4-8, 7-12
[Help] 2-11, 4-5, 6-17, 7-6
help information 2-11
[Insert Record] 5-13, 5-46, 6-10, 7-6
[Insert Record] function key 6-14
[Insert/Replace] 4-5, 7-6
keyboard mapping 4-2, 7-2, 12-9
[Last Line] 7-6
[Left] 2-4, 3-2, 4-5, 7-6
[List] 5-9, 6-12 to 6-13, 7-6
[Main Menu] 3-2, 4-5
[Menu] 6-18, 7-7
[Navigate] 7-7, 8-6, 8-13, 8-22
navigating functions 3-2
[Next Field] 2-2, 4-5, 5-46, 6-10, 7-7
[Next Record] 6-11, 6-16, 7-7
[Paste] 7-8
[Previous Field] 2-8, 3-2, 4-6, 7-8

[Previous Menu] 3-2, 4-6
[Previous Record] 6-11, 6-16, 7-8
[Print] 4-6, 7-8
[Quit] 2-3, 3-2, 4-6, 7-8
[Redefine Usr/Pwd] 4-6
[Refresh] 4-6, 7-9
[Right] 2-4, 3-2, 4-7, 7-9
[Scroll Down] 6-18 to 6-19, 7-9
[Scroll Left] 7-10
[Scroll Right] 7-10
[Scroll Up] 6-18 to 6-19, 7-10
[Search] 6-14, 7-10
[Select] 3-3, 4-7, 5-30, 5-45, 6-6, 6-8, 7-10
[Show BGM] 2-12, 3-2, 4-7
[Show Keys] 4-2, 4-7, 5-3, 7-2, 7-11
SQL*Forms [Exit/Cancel] 3-8
SQL*Forms function keys 4-2, 7-2
SQL*Menu (Design) 7-2
SQL*Menu (Run Menu) 4-2
typographic conventions iii
[Up] 3-7, 4-7, 6-11, 6-16, 7-11
user assistance functions 4-8
[Where Display] 3-9, 4-7
[Zoom In] 5-14, 5-16, 7-11, 8-10
[Zoom Out] 5-16, 7-11, 8-6, 8-10

G

-g and *-gx* switches 5-39, 12-6
Generate
  error messages B-14
  Generate before Executing Menu option 5-25, 5-29
  item in Action menu 12-6
  save before generate 5-24
Generate before Executing Menu option 5-25, 5-29
Generate item
  in Action submenu 5-25, 12-6
Generate Menu Application dialog box 5-25
Generating a menu application 5-25
GENMENU command
  -? switch 12-7
  -p switch B-15
  error messages B-14
  -g switches 5-39, 12-6
  -r switch 12-7
  SQL*Menu (Generate) 12-6
  switches B-14
Global variable 8-18, 9-6
  naming conventions 9-7
Grant access
  item in Admin submenu 5-40 to 5-41, 12-6
  roles 8-15
Grant Access to User dialog box 5-40 to 5-41
  equivalent switches 12-6 to 12-7
GRANT privilege 5-39, 12-7
Grant Role Access button
  Item Role window 5-15 to 5-16, 8-15

H

Help
  application help information 2-11, 12-8
  bookmark 6-19
  context-sensitive help 2-10
  function keys 2-11
  history 6-19
  index 6-19
  item help information 8-15
  key words 6-18
  message line 2-10
  online help system 6-17
  operator errors 2-10
  SQL*Menu (Design) 5-6, 6-17
  SQL*Menu (Run Menu) 2-10
[Help] function key 4-5, 7-6, 11-3
  help text for item 2-11, 8-15
  in the design interface 6-17
Help item 6-17
  SQL*Menu (Design) 5-6

Help submenu
SQL*Menu (Design) 5-6
Help system item 6-17 to 6-18
Help submenu 5-7
Help text
menu item 5-16, 8-15
omit from summary 12-8
scroll region 12-11
Hierarchy
application object 8-5
context 6-2, 8-2
menu item object 8-12
menu object 8-9
menu tree 1-4, 8-16
objects 5-2, 6-2, 7-11
parameter object 8-21
parent menu 1-2, 3-7, 3-9, 6-3
procedure object 8-26
Where option 3-9
Hint field 8-24
parameter 12-12
Hint line
See Message line
Horizontal scrolling 5-43, 6-16
Host item 5-36

I

-i switch 8-18
Identification field 1-6, 5-11, 8-7
IFORM state 10-3
Ins status line item 1-7, 4-5, 6-5, 11-5
INSERT privilege 5-39, 12-6 to 12-7
[Insert Record] function key 7-6
in a form or spread table 5-13, 6-10, 6-14
in a window 5-16, 5-46
[Insert/Replace] function key 4-5, 7-6, 11-3
Installation
linking SQL*Forms 8-17
SQL*Menu 1-10
Interface 6-1, G-2
See also Design interface
Interface element
alert 6-5
button 6-6, G-1
check box 6-8, G-1
data entry 6-5
dialog box 6-8
field 6-9
form 6-10
item 6-4
list of values 6-11
menu 6-3
message line 6-3 to 6-4
online help system 6-17
radio button 6-7
scroll region 6-13
spread table 6-14
status line 6-3 to 6-4
submenu 6-3
window G-3
Invoking a submenu
SQL*Menu command type 1 8-16
Invoking packaged procedures
PL/SQL block 9-6
SQL*Foms 10-15
Invoking SQL*Forms
packaged procedures 10-15
SQL*Menu command type 4 8-17, 11-7
Invoking SQL*Menu
DOCMENU syntax 12-8
full-screen display style 3-8, 12-5
GENMENU syntax 12-6
REPLACE_MENU 10-16
RUNMENU syntax 3-3, 12-4
SQLMENU syntax 12-4
Invoking SQL*Plus
SQL*Menu command type 5 8-18
Invoking SQL*ReportWriter
SQL*Menu command types 2 and 3 8-17

Item
command line 5-16, 10-15
command types 8-16
current item 2-4
default number 8-15
defining 5-14, 8-12
defining help text 5-16
definition 6-4
disabling 10-7
enabling 10-8
executing SQL*Forms procedure 10-15
interface element 6-4
invoking a procedure 5-22, 10-15
item text 5-15, 8-15
See also Menu item
name 3-3, 5-15, 8-3, 8-15, 9-10, B-22, B-32
number 3-7, 8-15
online help 2-11
parameter 5-19, 8-26
restrictions on selecting directly 3-9
selecting by cursor position 2-6, 3-3
selecting by letter 2-5, 3-3
selecting directly 3-7
short item name 3-3, 5-15, 8-15, B-22
unavailable 6-4, 8-14
unlimited number 12-9
Item Definition form and spread table 5-14, 5-19, 8-19 to 8-20, 12-11
command line 9-4
command types 11-7
error messages 9-10, B-22 to B-23, B-32
Item field 5-22
object definition 8-12
object name 9-10
reordering items 5-22
Short Item Name field 9-10, 12-11, B-22, B-32
Item parameter
creating 5-19
entering values 8-25
Item Role window 5-15, 5-42, 8-14 to 8-15, 12-13
Item text 8-15
Item Text field
full-screen menu 5-15
Item Text scroll region
column number 9-10
line number 9-10
ITEM_ENABLED packaged function 10-7 to 10-8

K

Keyboard map 4-2, 7-2
terminal definition 12-9
Keys
item in Help submenu 5-6

L

Language support A-3
[Last Line] function key 7-6
[Left] function key 4-5, 7-6, 11-3
in a bar menu 3-2
in a submenu 3-2
in an alert 6-6
in the application menu 2-4
LEFT macro 11-5
Library file
deleting 5-36
directory 8-7
DMM file 5-25, 8-3
generating 5-25, 5-34, 5-36, 5-38
naming conventions 8-3
save before generate 5-24
Limits
See Constraints
Line number
scroll region 9-10
Line number in a scroll region 9-10
[List] function key 5-9, 6-12 to 6-13, 7-6
List of values 6-11
Auto Restrict List of Values option 5-29
scrolling 6-12
search function 5-29, 6-12
Show List of Values option 5-29
List status line item 6-5, 6-13, 7-6
Local variable 9-6

Logging into SQL*Menu (Design) 5-3
Logging into SQL*Menu (Run Menu) 2-2
Login G-2
  database 11-7
  operating system 2-2, 5-3
  OPS$ usage 12-2
  remote 12-2
  SQL*Forms 8-17 to 8-18
  SQL*Plus 8-18
Login screen
  keyboard mapping 4-2, 7-2
  SQL*Menu (Design) 5-3, 12-4
  SQL*Menu (Run Menu) 2-2, 12-5
  using a remote database 12-3
LOGIN state 10-3

M

Macro command 11-2, G-2
  arguments 11-6
  as menu item 8-19
  corresponding function keys 11-2
  corresponding procedures 11-2
  failure 11-4
  processing 11-4
  restrictions on use 11-4
  SQL*Menu command type 6 8-19
  suppressing screen output 11-5
  syntax 11-2, 11-6
  upgrading from Version 4.1 8-29, 11-2
  valid states 10-3, 10-5 to 10-15
  without packaged procedure 11-5
Main menu 1-2, G-2
  display styles 3-3
  name 8-11
  Orders Application 2-3
  root menu error message B-10
  SQL*Menu (Design) 5-4
[Main Menu] function key 3-2, 4-5, 11-3
MAIN_MENU
  packaged procedure 10-9
Mapping function keys 4-2, 7-2, 12-9
Memory
  error messages B-30
  global variable 9-7
Menu 1-2, 6-3, G-2
  application menu 2-3
  background menu 2-12
  bar style 3-3, 3-5, 12-5
  command types 8-16
  common features 1-6
  defining 5-12, 8-9, 8-11
  display styles 3-3, 5-26, 5-29, 8-11, 12-5
  dynamic 1-4 to 1-5
  full-screen style 2-4, 3-6, 5-15, 8-25, 11-4, 12-5
  interface element 6-3
  item in Menu submenu 5-12
  limited number of menus 12-9
  main menu 1-2, 5-13, B-10
  menu bar 3-3
  menu parameter 5-19, 8-25, 10-14
  menu tree 1-2, 1-7, 8-16
  message line 1-6, 2-10
  names 3-7, 3-9, 5-18 to 5-19, 8-3, 8-11
  options 2-12, 2-14
  parent 1-2, 1-8, 3-7, 3-9, 6-3
  pull-down style 3-3 to 3-4, 12-5
  restrictions on selecting directly 3-9
  selecting items 2-4, 2-6, 3-7
  SQL*Menu states 11-4
  starting menu 1-8, 10-16
  status line 1-7, 2-12, 2-14, 5-14
  submenu defined 1-2
Menu application
  See Application
Menu bar
  bar style submenus 3-5
  menu display styles 3-3
Menu Definition form and spread table 5-12, 12-11
  error messages B-20 to B-21
  object definition 8-9
[Menu] function key 7-7
  in the help system 6-18 to 6-19

Menu item G-2
creating 5-14, 8-12
disabling 10-7
enabling 10-8
in main menu 5-14
in Menu submenu 5-12
See also Item
selecting 2-4
SQL*Menu (Design) 5-10, 5-12, 5-17
SQL*Menu functions 1-4
unavailable until select application 5-10, 5-12
Menu Name window 8-24
error messages B-21
Parameter Definition 5-18 to 5-19, 8-23
Menu parameter
creating 5-19
full-screen menus 8-25
processing 8-26
Menu path 3-9, 4-7
See also Where option
MENU state 10-3
Menu structure
See Menu tree
Menu submenu 5-10
Application item 5-10
Item item 5-14
Menu item 5-12
Parameter item 5-17
SQL*Menu (Design) 5-10, 5-12
Menu tree 1-2, 1-7, G-2
branch 3-7, 8-11, 8-16, G-1
hierarchy 1-4, 3-9
navigating 3-2, 3-7
nesting 1-8, 12-9
path 3-2, 3-9, 4-7
starting menu name 10-16
Where option 3-9, 8-11
MENU_CLEAR_FIELD
packaged procedure 10-9
MENU_FAILURE packaged function 10-3
MENU_FATAL packaged function 10-3
MENU_HELP
packaged procedure 10-10
MENU_MESSAGE
packaged procedure 10-10
MENU_NEXT_FIELD
packaged procedure 10-10
MENU_PARAMETER
packaged procedure 10-11
MENU_PREVIOUS_FIELD
packaged procedure 10-11
MENU_REDISPLAY
packaged procedure 10-11
MENU_SHOW_KEYS
packaged procedure 10-12
MENU_SUCCESS packaged function 10-3
Message line 1-6, 2-10, 6-4
error messages 2-10, 6-17
hints 2-10, 3-3, 6-3, 10-10
interface element 6-3 to 6-4
item text 5-15
pull-down and bar menus 3-3, 5-15
Show Detailed Working Messages option 5-29
Suppress Hints option 5-29, 6-4
working message 5-24, 5-34, 5-38, 5-41
Messages
typography iii
working message 5-24, 5-34, 5-38, 5-41
MFORM state 10-3
MNU-xxxxx
See Error message
Mnu: status line item 6-5
Modifying application information 5-10
Moving an application 5-37
See also Exporting
Multi-user system 1-9, G-2

N

Name
application 8-6, 12-5 to 12-6, 12-8
field in Export dialog box 5-37
field in New Menu Application dialog box 5-9
item 3-3, 5-15, 8-15, B-22
main menu 8-11
menu 8-11
new application 5-9
procedure 9-5

NAME_IN packaged function
PL/SQL arguments 10-5
Naming conventions
error messages B-19 to B-23, B-25, B-27, B-29 to B-30
main menu 5-13
menus 8-3
See also Object
procedure 8-28, 9-5
variables 9-7
National Language Support A-3
[Navigate] function key 7-7, 8-6, 8-10, 8-13, 8-22, 8-28
Navigation
direct selection 3-7
function keys 3-2
in full-screen menus 1-4, 3-7, 8-11
in SQL*Menu (Run Menu) 1-4, 3-2
menu hierarchy 1-4
Nesting 1-8, G-2
limitations 12-9
See also Menu tree
New item 5-8
New menu application
creating 5-8
error messages B-27
library file 5-25, 5-34
New Menu Application dialog box 5-8 to 5-9
NEW_APPLICATION
packaged procedure 10-12
NEW_USER
packaged procedure 10-12
NEWAPL macro command 11-6
NEWUSER macro command 11-7
[Next Field] function key 4-5, 7-7, 11-3
in a dialog box 2-8, 5-34 to 5-36, 5-38
in a form or spread table 5-45, 6-10, 6-14
in a list of values 6-12
in a menu 3-2
in an alert 5-46
in the login screen 2-2, 5-3
[Next Record] function key 7-7
in a form or spread table 6-11, 6-16
NEXT_MENU_ITEM
packaged procedure 10-13
NLS*WorkBench A-3
No button 5-5, 5-45 to 5-46
NODISP macro 11-5
NOT NULL 12-10 to 12-14
NOTE alert label 6-5
NULL 10-9, 12-5, 12-10, G-2
directory 8-7
parameter value 8-24

O

Object 8-2, G-2
application 5-10, 8-5
context 6-2
copying 5-23
documenting references 5-28, 12-8
hierarchy 5-2, 6-2, 7-11, 8-2
item 8-12
menu 5-12, 8-9
menu item 5-14
naming conventions 5-19, 8-3, 9-7, B-27, B-29
parameter 5-17, 8-21
PL/SQL reference 9-6
procedure 5-21, 8-26
record 6-10, 6-14
reference 5-23, 9-6, 9-8, 12-14
Object definition
application 5-10 to 5-11, 8-5, 12-10
copying 5-23
documenting an application 5-27 to 5-28, 12-8
documenting references 5-28, 12-8
item 5-14, 5-22, 8-19, 9-4, 12-11, B-22, B-32
item name 9-10, 12-11, B-22, B-32
menu 5-12, 8-9, 12-11
menu item 8-12
parameter 5-17, 8-21, 12-12
procedure 5-21, 8-26, 8-29, 9-4, 9-9, 12-14
referencing 5-23
role 5-42, 5-45, 12-13
Object hierarchy
application 8-5
menu 8-9
menu item 8-12
parameter 8-21
procedure 8-26

Object ref item
equivalent switch 12-8
in Print Doc submenu 5-28
Online help system 2-11, 5-6, 6-17
bookmark 6-19
contents 6-18
history 6-19
index 6-19
key words 6-18
scrolling 6-18
Open item
Action submenu 5-9
Open Menu Application dialog box 5-9
Opening a menu application 5-9
Operating system G-3
commands 1-4, 2-14, 5-36, 5-44, 8-17
GENMENU command 5-39
logging in 2-2
menu nesting limits 12-9
multi-user 1-9
OPS$ login 12-2
OSC option 2-14, 5-44, 12-13
prompt 2-14, 3-8, 12-4 to 12-5, 12-7 to 12-8, G-3
running SQL*Menu 1-3, 2-2, 5-3, 5-39, 12-2 to 12-3
Operations
application maintenance 5-33
basic design operations 5-6
security 5-38
setting options in SQL*Menu (Design) 5-29
Operator ii, 1-2
assigning to roles 5-45
deleting from role 5-46
function keys 4-2
privilege level 5-39, 12-6
OPS$ login 12-2
Option
Background Menu option 2-12, 5-44
Debug Mode option 5-26, 5-44
menu display style 5-26, 5-29
See also Option Selection form
OS Command option 2-14, 5-44
SQL*Menu (Design) 5-29
SQL*Menu (Run Menu) 1-7
Where option 3-9, 4-7
Option Selection form
Auto Restrict List of Values option 5-29, 6-13
Disable PL/SQL Compilation option 5-29
Generate before Executing Menu option 5-25, 5-29
menu display style 5-26, 5-29
radio buttons 5-29, 6-7
Show Detailed Working Messages option 5-29
Show List of Values option 5-29, 6-13
Suppress Hints option 5-29, 6-4
Use Forms as Default option 5-29, 6-11, 6-16
Options 5-29, G-3
application 12-13
item in main menu 5-30, 6-4
presetting a design mapping 5-32
presetting a menu style 5-32
presetting design options 5-31
setting options 5-29
SQL*Menu (Generate) B-14
SQL*Menu (Run Menu) 5-42, 12-13
user preference file 5-31
Options item
SQL*Menu (Design) 5-30, 6-4
ORACLE
naming conventions 5-21, 8-3, 8-28, 9-5, 9-7
reserved words 8-3
username 5-41, 12-4 to 12-6, 12-8
username and password 1-9, 2-2, 5-3, 11-7
ORACLE Installation and User's Guide 1-8
Oracle*Terminal 12-3 to 12-5, 12-9
administration 12-9
Orders Application
exiting 2-3
invoking full-screen style 3-8 to 3-9
logging in 2-2
menu tree 1-7
OSCMD macro 8-20
packaged procedure 8-20
report parameters 8-19
sample 1-5
sample order-entry form 3-8
search string parameter 8-17
OS command
See Operating system
option 2-14

OS Command check box 5-44
OS Command option 5-44, 12-13
OS_COMMAND
  packaged procedure 10-13
OS_COMMANDl
  packaged procedure 10-13
OSC
  See OS Command option
OSC status line item 1-7, 2-14
  See also Operating system
OSCMD macro command 8-20, 11-7

P

Packaged function
  ITEM_ENABLED 10-7 to 10-8
  NAME_IN for indirect reference 10-5
  packaged procedure execution 10-3
  PL/SQL 9-10
Packaged procedure 8-26, 10-2
  arguments 10-4
  corresponding macros 11-2
  definitions 10-5
  ERASE 9-7
  failure 9-10, 10-2
  in PL/SQL 9-6
  macros without packaged procedures 11-5
  PL/SQL 8-20
  processing 10-2
  restrictions on use 10-2
  sample 8-20
  SQL*Forms 8-20, 10-15
  testing execution 10-3
  using 8-28
  valid states 10-3, 10-5 to 10-15
Parameter G-3
  field 8-24
  item in Menu submenu 5-17
  See also Substitution parameter
Parameter Definition form and spread table 5-17, 12-12
  error messages B-21, B-23 to B-25
  object definition 8-21
Parent menu
  See Hierarchy
Password G-3
  -z switch 12-5
  changing user 11-7
  security 12-5
  SQL*Forms authorization 8-18, 11-7
  SQL*Menu authorization 12-4 to 12-6, 12-8
  stored internally 12-5
  system authorization 1-9
[Paste] function key 7-8
PL/SQL 1-5, 9-2
  as menu item 8-20
  block 1-5, 9-3 to 9-4
  compilation 5-29, 9-8
  Compilation Errors form 9-9
  error messages 9-9, B-29, B-31
  exception handling 9-11
  executable statement failure 9-10
  local variable 9-6, 10-5
  packaged procedures 9-6, 10-2, 10-15
  procedure 5-21, 12-14
  reference restrictions 9-8
  referencing objects 5-17, 9-6
  SQL*Menu command type 7 8-20
  substitution parameters 10-14
  unnamed procedure 5-22
  using substitution parameters 5-17, 8-25
  variables 9-6
Planning an application 5-2
Predefine parameter
  See Substitution parameter
Preference file
  See User preference file
[Previous Field] function key 4-6, 7-8, 11-3
  in a dialog box 2-8, 3-2
  in a menu 3-2
[Previous Menu] function key 3-2, 4-6, 11-3
[Previous Record] function key 7-8
  in a form or spread table 6-11, 6-16
PREVIOUS_MENU
  packaged procedure 10-14
PREVIOUS_MENU_ITEM
  packaged procedure 10-14

Print doc
error messages B-12
item in Action submenu 5-27, 12-8
Print Documentation dialog box 5-27
Print Documentation submenu 5-27
[Print] function key 4-6, 7-8
Printing object information 5-27, 12-8, B-12
Privilege
Grant Access to User dialog box 5-41, 12-6 to 12-7
menu item 12-13
Privilege Level field 5-41
user 5-39
Procedure 8-26, 9-4
base table 12-14
creating 5-21, 8-26, 8-28
definition 8-26
Disable PL/SQL Compilation option 5-29
error messages B-29 to B-30
invoked by item 5-22, 8-20
item in main menu 5-21, 8-27
naming conventions 8-28, 9-5
packaged 8-26, 10-5
See also Packaged procedure
See also PL/SQL
PL/SQL block 5-22, 9-4
Procedure Definition form and spread table 5-21
specification 9-9
syntax 9-5
unnamed 5-22
using 5-22, 8-28
Procedure Definition form and spread table 5-21, 8-29, 9-4, 12-14
compiling a procedure 9-9
error messages B-30
object definition 8-26
Procedure Name field 8-28
Procedure Text scroll region
column number 9-10
line number 9-10
online help 6-17
PL/SQL 8-28
syntax 9-5
Prompt G-3
See also Message line
operating system 2-2, 2-14
substitution parameter 8-24 to 8-25
Pull-down menu 3-4
application identification 8-7
invoking 12-5
item text 8-15
navigating 3-2
selecting an item 2-4
short item name 8-15
using 3-3
Purpose scroll region 8-11, 12-11
&PW substitution parameter 8-25, 11-6, 12-5
with automatic login 12-2

Q

QUERY_PARAMETER
packaged procedure 10-2, 10-14
Quiet mode
RUNMENU command 12-5
[Quit] function key 2-3, 3-2, 4-6, 7-8, 11-3
Quit item 5-4
Quitting
SQL*Menu (Design) 5-4
SQL*Menu (Run Menu) 2-3

R

-r switch 8-18
Radio button 6-6 to 6-7
See also Button
in Option Selection form 5-30
Record 5-12, G-3
in a form 6-10
in a spread table 5-12, 5-14, 6-14
[Redefine Usr/Pwd] function key 4-6, 11-3
REDISP macro 11-5 to 11-6
Reference
See [Copy Object] function key
object 12-14
PL/SQL 9-6

[Refresh] function key 4-6, 7-9, 11-3
Release date 8-8
Remote login 12-2
Rename item 5-35
Rename menu application
  error messages B-27
Rename Menu Application dialog box 5-35
Renaming an application 5-35
Rep status line item 1-7, 4-5, 6-5, 11-5
REPLACE_MENU
  SQL*Forms packaged procedure 10-16
Report run from SQL*Menu 2-8
Required check box
  substitution parameter 8-24
Requirements 1-8
Reserved words 8-3
Restrictions
  See Constraints
Revoking access to SQL*Menu 5-41, 12-7
[Right] function key 4-7, 7-9, 11-3
  in a bar menu 3-2
  in a submenu 3-2
  in an alert 6-6
  in the application menu 2-4
RIGHT macro 11-5
Role 1-5, 5-42
  access to items 5-16, 8-15
  assigning users 5-45
  creating 5-2, 5-38, 5-42
  deleting 5-44
  deleting users 5-46
  error messages B-25 to B-26, B-28 to B-29
  menu called by SQL*Forms 10-16
  See also Security
  work-class conversion 5-42, A-2
Role Definition form and spread table 5-42, 5-45, 12-13
  error messages B-25
Root menu 1-2
  called by SQL*Forms 1-8, 10-16
  error messages B-10
  See also Main menu
RUNMENU command
  -? switch 12-5
  error messages B-3
  SQL*Menu (Run Menu) 12-4
  switches 3-3, 3-8

S

Sample application
  exiting 2-3
  help information 2-11
  invoking full-screen style 3-8 to 3-9
  menu tree 1-7
  order-entry form 3-8
  Orders Application 1-5
  OSCMD macro 8-20
  packaged procedure 8-20
  report parameters 8-19
  requirements for use 1-9
  ruining a report 2-8
  search string parameter 8-17
Save item 5-24
Save Menu Application dialog box 5-24
Saving a menu application 5-24
  deleting records 7-5
  error messages B-27
Scroll bar
  arrow indicators 6-11, 6-16
  horizontal scroll bar 6-16
  vertical scroll bar 6-12, 6-16, 6-18
[Scroll Down] function key 6-18 to 6-19, 7-9
[Scroll Left] function key 7-10
Scroll region 6-13
  Command Line 5-22, 8-14, 8-16 to 8-20, 9-4
  Comments 5-44, 6-13, 12-13
  Help Text 12-11
  Item Text 9-10
  line number 9-10
  Procedure Text 6-17, 9-10
  Purpose 8-11, 12-11
  search function 6-14
[SCROLL Right] function key 7-10
[Scroll Up] function key 6-18 to 6-19, 7-10

Scrolling
list of values 6-12
menu items 12-9
online help system 6-18
scroll region 6-13
spread table 5-43, 6-16
SQL*Menu (Run Menu) 3-3, 12-9
Search dialog box 6-14
Search function
directory listing 8-17
list of values 5-29, 6-12
scroll region 6-14
[Search] function key 7-10
in a scroll region 6-14
Security
application level 1-4
creating roles 5-2, 5-42
error messages B-25 to B-26, B-28 to B-29
exporting applications 5-38
item in Admin submenu 5-42, 5-45
menu level 1-4
operations 5-38
password storage 12-5
revoking access 5-41, 12-7
See also Role
saving security table 5-44, 5-46
SQL*Forms 10-16
upgrading applications A-2
[Select] function key 4-7, 7-10
in a check box 5-18, 5-30, 6-8
in a full-screen menu 3-7
in a list of values 6-12
in a menu 2-4, 3-3
in an alert 5-5, 5-46, 6-6
on a button 5-18 to 5-19, 5-45, 6-6
on a radio button 5-30, 6-7
Select Menus button 5-19
Menu Name window 8-24
SELECT privilege 5-39, 12-6 to 12-7
Select Users button 5-45
Selecting an item 2-3
by cursor position 2-6
by letter 2-3, 2-5
in full-screen menus 3-7
in pull-down menus 2-4, 3-3
Send to Printer check box 5-28
Setting options
SQL*Menu (Design) 5-29
Short application name 8-8
Short item name
field B-22, B-32
pull-down and bar menus 3-3, 5-15, 8-15
Short Item Name field 9-10, 12-11, B-32
See also Item Definition form and spread table
[Show BGM] function key 2-12, 3-2, 4-7, 11-3
Show Detailed Working Messages option 5-29
[Show Keys] function key 4-7, 7-2, 7-11, 11-3
getting help 2-11, 4-2, 5-3
Show List of Values option 5-29, 6-13
SHOW_BACKGROUND_MENU
packaged procedure 10-15
Single-user system 1-10, G-3
See also Multi-user system
&SO substitution parameter 8-25
assigning a value 11-6
Software
used with SQL*Menu 1-2
Software used with SQL*Menu 1-8
Specification
See also Procedure
Specification of a procedure 9-9
Spread table 6-14
context 6-15
display style 5-12, 5-14, 5-29, 5-42, 7-11
Item Definition 5-14
Menu Definition 5-12
Parameter Definition 5-17
Procedure Definition 5-21, 8-26, 8-29
Role Definition 5-42, 5-45
scrolling 6-16

SQL 1-5, 9-2
  export file 5-38, 12-6
SQL statement
  failure 9-11
SQL*Forms
  global variables 8-18
  integration with SQL*Menu 1-4, 8-17 to 8-18, 9-7
  invoking with SQL*Menu 8-17, 8-25, 11-7
  ORDERS sample form 1-5
  packaged procedures 9-8, 10-15
  referencing objects 9-6, 9-8
  REPLACE_MENU 10-16
  required version 1-8
  starting menu 1-8
  switches 8-18
SQL*Menu
  -? switch 12-3
  administrators ii, 1-2, 5-39, 5-45
  authorization to use 1-9
  base tables 5-39, 12-10
  command types 8-16
  components 1-3, 12-3
  constraints 12-9
  context 10-2, 11-4
  deleting roles 5-44
  designers ii, 1-2, 5-39
  DOCMENU command 12-3
  document file 5-28, 12-8
  dropping users 5-41
  dynamic menu approach 1-5
  enrolling users 5-38
  enrollment 1-5, 1-9
  features 1-3
  function keys 4-2, 7-2
  GENMENU command 12-3
  granting access 5-39, 12-7
  help information 2-10
  installation 1-10
  integrating software products 1-2, 1-4
  interface elements 6-1
  menu items 1-4
  names 8-3
  navigating 3-2
  online help system 6-17
  operators ii, 1-2, 5-39
  remote database 12-3
  requirements 1-8
  roles 5-16, 5-42, 5-45 to 5-46
  RUNMENU command 12-3 to 12-4
  security 5-42
  SQLMENU command 5-3
  starting up 2-2, 5-3
  states 10-2 to 10-3, 10-5 to 10-15, 11-4
  unavailable items 6-4
  upgrading A-2
  views of base tables 12-10
SQL*Menu (Design)
  -? switch 12-4
  -c switch 12-4
  -e switch 12-4
  -r switch 12-4
  -w switch 12-4
  application maintenance operations 5-33
  basic operations 5-6
  command line for an item 5-16
  context 6-2
  copying an application 5-33
  creating a new application 5-8
  creating roles 5-42
  defining a menu 5-12
  defining a menu item 5-14
  defining a parameter 5-17
  defining a procedure 5-21
  deleting an application 5-36
  error messages B-19
  executing an application 5-26

function keys 7-2
generating an application 5-25
Help submenu 5-6
help text for an item 5-16
interface elements 6-1, 6-5
item 6-4
logging in 5-3
main menu 5-4
menu 6-3
massage line 6-4
modifying an application definition 5-10
moving an application 5-37
online help system 5-6, 6-17
oopening an application 5-9
options 5-25 to 5-26
overview 5-2
planning an application 5-2
quitting 5-4
renaming an application 5-35
saving an application 5-24
security operations 5-38
setting options 5-29
SQL*Menu command 5-3, 12-4
status line 6-4
submenu 6-3
Suppress Hints option 6-4
terminal definition 12-4
unavailable items 6-4
unloading an application 5-37
SQL*Menu (Document)
-? switch 12-8
-b switch 12-8
-f switch 12-8
-o switch 12-8
command syntax 12-8
DOCMENU command 12-8
error message B-12
SQL*Menu (Generate)
-? switch 12-7
-e switch 12-6
command syntax 12-6
error messages B-14
-g switches 12-6
GENMENU command 12-6
-r switch 12-7
SQL*Menu (Run Menu)
-? switch 12-5
-c switch 12-5
-e switch 12-5
-m switch 12-5
-q switch 12-5
-r switch 12-5
-w switch 12-5
-z switch 12-5
background menu 2-12
command syntax 12-4
context 10-2, 11-4
error messages B-3
exiting 2-3
function keys 4-2
help information 2-10
logging in 2-2, 3-3, 3-8 to 3-9
menu display styles 3-3
navigating 3-2, 3-7
options 1-7, 2-12, 2-14, 5-26, 5-42, 12-13
parameter values 2-8
RUNMENU command 3-3, 3-8, 12-4
sample application 2-2
SQL*Menu states 11-4
states 10-2 to 10-3, 10-5 to 10-15
suppressing password 12-5
syntax 3-3
terminal definition 12-5
Where option 3-9, 4-7

SQL*Menu DBA
See also Database administrator
privileges 5-39
responsibilities 5-45
SQL*Net
using with SQL*Menu 12-2
SQL*Plus
invoking from SQL*Menu 8-18
moving an application 5-38
required version 1-8
SQL*ReportWriter
invoking with SQL*Menu 8-17
SQLMENU command 12-4
-? switch 12-4
SQLMENU.CFG file 5-31
Starting menu
called by SQL*Forms 1-8, 10-16
REPLACE_MENU packaged procedure 10-16
Starting up SQL*Menu 2-2, 5-3, 12-3
Status line 1-7, 5-14
App: item 6-5
BGM option 2-12
current menu 5-14
insert/replace indicators 4-5, 6-5, 11-5
interface element 6-3 to 6-4
List option 6-13
menu names 3-7, 8-11
Mnu: item 6-5
OSC option 2-14
STOP alert label 6-5
String G-3
Submenu 1-8
bar style 3-5
display styles 3-3
first-level 3-4 to 3-6
full-screen 3-6
interface element 6-3
invocation 8-16
navigating 3-2
pull-down 3-4
second-level 3-4 to 3-6
submenu indicator 2-5
Substitution parameter 8-25, G-3
&AD 8-25, 11-6
base table 12-12
default value 8-23
defining 5-17, 8-21
definition 8-25
entering values 2-8, 8-25
hint 8-24
item parameter 5-19, 8-26
menu and main menu parameters 5-19, 8-26
menu parameter 8-24 to 8-25, 10-14
Orders Application 8-17, 8-19
Parameter Definition form and spread table 5-17
parameter substitution 1-4, 12-5
passed to SQL*Forms 8-17
passed to SQL*Plus 8-18
predefine parameters 5-17, 8-25, 11-6
prompt 8-24
&PW 8-25, 11-6, 12-2, 12-5
size 8-24
&SO 8-25, 11-6
&TT 8-25, 11-6
&UN 8-25, 11-6
Subtitle in full-screen menus 8-11
Summary item
equivalent switch 12-8
in Print Doc submenu 5-28
Suppress Hints option 5-29, 6-4
SUSPEND macro 11-5
Switch G-3
-? switch 12-3
-c mapping 12-4 to 12-5
-m 3-3, 3-8 to 39
-m *fullscreen* 12-5
-c 8-18
DOCMENU 12-8
documentation 12-8
-e 8-18
-g -ge$_t$-gd -ga 5-39, 12-6
GENMENU 12-6, B-14 to B-15
-i 8-18

-r 8-18
RUNMENU 3-3, 3-8, 12-5
SQL*Forms 8-18
SQLMENU 12-4
-w 8-18
Syntax G-3
-? switch 12-3
argument of macro 11-6
argument of packaged procedure 10-4
compilation error messages 9-10
DOCMENU 12-8
GENMENU 12-6
macros 11-2, 11-6
parameter 8-25
PL/SQL reference 9-6
procedure 9-5
RUNMENU 3-3, 3-8 to 3-9, 12-4
SQL*Menu packaged procedures 10-5
SQLMENU 12-4
typography iii
System
See Operating system
System components 1-3
System variable 9-7

T

Table
base tables 5-39, 12-6, 12-10
security table 5-38, 5-44, 5-46
See also Spread table
Terminal G-3
Terminal definition
Oracle*Terminal 12-9
RUNMENU command 12-5
SQLMENU command 12-4
&TT substitution parameter 8-25, 11-6
TERMINATE
packaged procedure 10-15
Title
application title bar 1-6
full-screen menus 8-12
Tree
See Menu tree
&TT substitution parameter 8-25, 11-6
Type
See Command types
Typographic conventions iii

U

&UN substitution parameter 8-25
with automatic login 12-2
Unload item
in Action submenu 5-37, 12-6
Unloading an application 5-37
See also Exporting
Unnamed procedure 5-22, 8-20
See also Anonymous block
[Up] function key 4-7, 7-11, 11-3
in a form or spread table 5-20, 6-11, 6-16
in a full-screen menu 2-4, 3-7
in a list of values 6-12
in **a** pull-down menu 3-2
UPDATE privilege 5-39, 12-6 to 12-7
Upgrading from Version 4.1
macros 11-2
security A-2
work-class conversion 5-42, A-2
Upper Case check box 8-25
Use Forms as Default option 5-29, 6-11, 6-16
User preference file 5-31
Username 12-4, G-3
changing user 11-7
designer 5-3
SCOTT 2-2, 3-8 to 3-9, 11-7
SQL*Forms 8-18
SQL*Menu authorization 12-4 to 12-6, 12-8
system authorization 1-9
window B-26
Username window
Role Definition 5-45, 12-13

Users
administrators 5-39
assigning to roles 5-45
deleting from role 5-46
designers 5-39, 12-6
enrollment 5-38
operators 5-39, 12-6
revoking access 5-41, 12-7

V

Validation 6-10
Validation error message
Application Definition form B-19 to B-20
Item Definition B-22 to B-23
Menu Definition B-20 to B-21
MNU-20xxx B-19
Parameter Definition B-21, B-23 to B-25
Procedure Definition B-30
Role Definition B-25
Variable
global 9-6
local 9-6
PL/SQL 9-2, 9-6
substitution parameter 8-26
system 9-7
Version
item in Help submenu 5-6
release date 8-8
Version Number field 8-8
Vertical scrolling 6-12, 6-16, 6-18
Views of base tables 12-10

W

-w switch 8-18
[Where Display] function key 3-9, 4-7, 11-3
Where option
example 3-10
finding menu location 3-9
menu names 3-7, 8-11
[Where Display] function key 4-7
WHERE_DISPLAY
packaged procedure 10-15
Window G-3
alert 6-5
Item Role window 5-15, 5-42, 8-14, 12-13
Menu Name window 5-18 to 5-19, 8-23 to 8-24
Username window 5-45, 12-13
Work-class conversion
See role
Working message 5-24, 5-34 to 5-36, 5-38, 5-41
Show Detailed Working Messages option 5-29

Y

Yes button 5-5, 5-45 to 5-46

Z

[Zoom In] function key 5-14, 5-16, 7-11, 8-10
[Zoom Out] function key 5-16, 7-11, 8-6, 8-10