

iSQL*Plus

User's Guide and Reference

Release 9.0.1

June 2001

Part No. A88826-01

ORACLE®

iSQL*Plus User's Guide and Reference, Release 9.0.1

Part No. A88826-01

Copyright © 1996, 2001, Oracle Corporation. All rights reserved.

Primary Author: Simon Watt

Contributors: Andrew Code, Alison Goggin, Alison Holloway, Christopher Jones, Anita Lam, Luan Nim, Andrei Souleimanian, Christopher Tan, Ian Wu.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle Net, Oracle HTTP Server, Oracle7, Oracle8, Oracle8i, Oracle9i, PL/SQL, iSQL*Plus and SQL*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	vii
Preface.....	ix
1 Introduction to <i>iSQL*Plus</i>	
What is <i>iSQL*Plus</i>?	1-2
<i>iSQL*Plus</i> Architecture	1-2
2 Configuring <i>iSQL*Plus</i>	
System Requirements	2-2
Installation	2-2
Configuring the Client Tier	2-3
Configuring the Middle Tier	2-4
Configuring the <i>iSQL*Plus</i> Server	2-4
Configuring the Oracle HTTP Server	2-7
Configuring the Database Tier.....	2-8
<i>iSQL*Plus</i> Security.....	2-8
Configuring Globalization Support	2-12
3 The <i>iSQL*Plus</i> User Interface	
The <i>iSQL*Plus</i> Log In Screen.....	3-2
The <i>iSQL*Plus</i> Work Screen.....	3-4
The Change Password Screen	3-6

4 Using *i*SQL*Plus

Running <i>i</i> SQL*Plus	4-2
Entering Statements	4-2
Loading Scripts.....	4-3
Entering Substitution Variables.....	4-4
Creating Dynamic Reports.....	4-6
Using SET and COLUMN Commands	4-9
Getting Help	4-9
Exiting <i>i</i> SQL*Plus.....	4-9

5 Command Reference

SQL*Plus Command Summary.....	5-2
@ (“at” sign).....	5-5
@@ (double “at” sign).....	5-7
/ (slash)	5-9
ARCHIVE LOG	5-10
ATTRIBUTE	5-13
BREAK	5-15
BTITLE	5-20
CLEAR.....	5-21
COLUMN	5-23
COMPUTE.....	5-34
CONNECT.....	5-40
COPY	5-42
DEFINE	5-46
DESCRIBE.....	5-48
DISCONNECT	5-54
EXECUTE.....	5-55
HELP	5-56
LIST	5-57
PRINT.....	5-59
PROMPT	5-61
RECOVER	5-62
REMARK	5-69
REPFOOTER.....	5-70

REPHEADER	5-72
RUN	5-75
SET	5-76
SHOW	5-96
SHUTDOWN	5-101
START	5-103
STARTUP	5-105
TIMING	5-108
TTITLE	5-110
UNDEFINE	5-114
VARIABLE	5-115
WHENEVER OSERROR	5-122
WHENEVER SQLERROR	5-123

A SQL*Plus Error Messages

B Security

C Unsupported SQL*Plus Commands

Glossary

Index

Send Us Your Comments

iSQL*Plus User's Guide and Reference, Release 9.0.1

Part No. A88826-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: sqlplus@oracle.com
- FAX: +61 3 9690 0043 Attn: SQL*Plus Documentation Manager
- Postal service:
SQL*Plus Documentation Manager
Australian Product Development Centre
Oracle Corporation Australia Pty Ltd
324 St Kilda Road,
Melbourne, VIC 3004,
Australia

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services. There is also an iSQL*Plus Discussion Forum on the Oracle Technology Network located at <http://technet.oracle.com/> which may be helpful for iSQL*Plus issues.

Preface

The *iSQL*Plus User's Guide and Reference* provides information about the iSQL*Plus user interface, and the Oracle HTTP Server and Oracle9i configuration required to use iSQL*Plus.

This preface contains these topics:

- Audience
- Organization
- Related Documentation
- Conventions
- Documentation Accessibility

Audience

*iSQL*Plus User's Guide and Reference* is intended for end users and system administrators who perform the following tasks:

- Enter, edit, run store, retrieve, and save SQL commands and PL/SQL blocks.
- Calculate, store, and print query results.
- List column definitions for any table.
- Access and copy data between databases.
- Perform database administration.

To use this document, you need a basic understanding of the SQL database language. If you do not have any familiarity with this database tool, you should refer to the *Oracle9i SQL Reference*. If you plan to use the PL/SQL database language in conjunction with SQL*Plus, refer to the *PL/SQL User's Guide and Reference* for information on using PL/SQL.

Organization

This document contains:

Chapter 1, "Introduction to iSQL*Plus"

Provides introductory information to help you get started with *iSQL*Plus*.

Chapter 2, "Configuring iSQL*Plus"

Explains how to configure your web browser, the Oracle HTTP Server and Oracle9i.

Chapter 3, "The iSQL*Plus User Interface"

Provides a description of the screens in the *iSQL*Plus* User Interface.

Chapter 4, "Using iSQL*Plus"

Explains how to use *iSQL*Plus* and its features.

Chapter 5, "Command Reference"

Provides a reference for commands specific to *iSQL*Plus*.

Appendix A, "SQL*Plus Error Messages"

Provides a list of SQL*Plus and iSQL*Plus Error Messages.

Appendix B, "Security"

Explains how to restrict access to certain SQL*Plus and SQL commands.

Appendix C, "Unsupported SQL*Plus Commands"

Provides a list of commands unsupported in this release of iSQL*Plus.

Glossary

Defines technical terms associated with Oracle and SQL*Plus.

Related Documentation

For more information, see these Oracle resources:

- The *SQL*Plus 9.0.1 Release Bulletin* containing SQL*Plus general release notes is located in %ORACLE_HOME%\sqlplus\doc\readme.doc.
- The *iSQL*Plus 9.0.1 Release Notes* containing specific iSQL*Plus release notes are located in %ORACLE_HOME%\sqlplus\admin\iplus\README.htm, or after installation, at <http://host.domain/iplus/README.htm>.
- *SQL*Plus User's Guide and Reference* (Generic Documentation CD ROM).
- *SQL*Plus Quick Reference* (Generic Documentation CD ROM).
- *SQL*Plus Getting Started for Windows* (Windows Documentation CD ROM).

Many of the examples in this book use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle9i Sample Schemas* for information on how these schemas were created and how you can use them yourself.

In North America, printed documentation is available for sale in the Oracle Store at

<http://oraclystore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

<http://www.oraclebookshop.com/>

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://technet.oracle.com/membership/index.htm>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://technet.oracle.com/docs/index.htm>

Whitepapers, sample code, frequently asked questions and other useful information is regularly posted to the SQL*Plus section on OTN at

http://technet.oracle.com/tech/sql_plus/

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples
- Conventions for Windows Operating Systems

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.

Convention	Meaning	Example
UPPERCASE monospace (fixed-width font)	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width font)	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the \disk1\oracle\dfs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.
lowercase monospace (fixed-width font) <i>italic</i>	Lowercase monospace italic font represents placeholders or variables.	You can specify the <i>managed_clause</i> . Run <i>old_release</i> .SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. If users are expected to type them into the system, they are identified by the keyboard icon shown in the margin following. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:



```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

Similarly, output from an example is identified by a computer screen icon in the margin as shown in the margin following.



```
PAGESIZE 24
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> That we have omitted parts of the code that are not directly related to the example That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;

Convention	Meaning	Example
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program. For example, to start Oracle Database Configuration Assistant, you must click the Start button on the taskbar and then choose Programs > Oracle - <i>HOME_NAME</i> > Database Administration > Database Configuration Assistant.	Choose Start > Programs > Oracle - <i>HOME_NAME</i> > Database Administration > Database Configuration Assistant
C:\>	Represents the Windows command prompt of the current hard disk drive. Your prompt reflects the subdirectory in which you are working. Referred to as the command prompt in this guide.	C:\oracle\oradata>
<i>HOME_NAME</i>	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	C:\> net start Oracle <i>HOME_NAME</i> TNSListener

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to 8.1, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default was:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin95 for Windows 95 ■ C:\orawin98 for Windows 98 <p>or whatever you called your Oracle home.</p> <p>In this Optimal Flexible Architecture (OFA)-compliant release, all subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install release 9.0 on a computer with no other Oracle software installed, the default setting for the first Oracle home directory is C:\oracle\ora90. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>See <i>Oracle9i Getting Started for Windows</i> for additional information on OFA compliances and for information on installing Oracle products in non-OFA compliant directories.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.

Documentation Accessibility

Oracle's goal is to make our products, services, and supporting documentation accessible to the disabled community with good usability. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Introduction to *i*SQL*Plus

This chapter provides introductory information about *i*SQL*Plus architecture.

Specific topics discussed are:

- What is *i*SQL*Plus?
- *i*SQL*Plus Architecture

What is iSQL*Plus?

iSQL*Plus is a browser-based interface to SQL*Plus. iSQL*Plus is a component of the SQL*Plus product.

iSQL*Plus enables you to use a web browser to connect to Oracle9i and perform the same actions as you would through the command line version of SQL*Plus (known as *SQL*Plus* in this guide). You can use iSQL*Plus to write SQL*Plus, SQL and PL/SQL commands to:

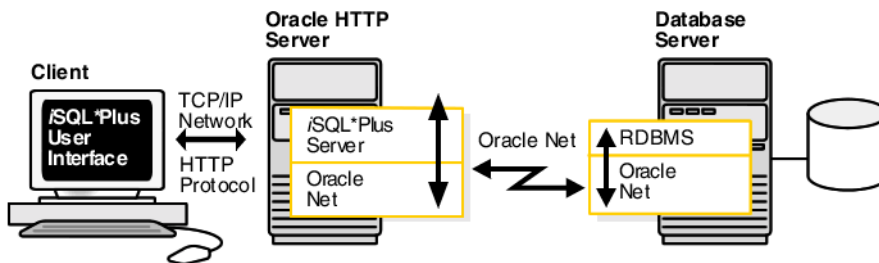
- Enter, edit, run and save SQL commands and PL/SQL blocks.
- Calculate, and print query results.
- List column definitions for any table.
- Access and copy data between databases.
- Perform database administration.

See Chapter 3, "The iSQL*Plus User Interface" for more information about the iSQL*Plus user interface.

iSQL*Plus Architecture

iSQL*Plus uses a three-tier architectural model comprising:

- Client Tier (iSQL*Plus user interface, typically a web browser).
- Middle Tier (iSQL*Plus Server, Oracle Net, and Oracle HTTP Server).
- Database Tier (Oracle9i).



The three tiers may or may not be on the same machine. The iSQL*Plus Server must be on the same machine as the Oracle HTTP Server. The middle tier coordinates interactions and resources between the client tier and the database tier. The

database tier is Oracle9i. Oracle9i can be installed and accessed on the middle tier, or accessed via Oracle Net.

iSQL*Plus User Interface

The iSQL*Plus user interface runs in a web browser connected to the Internet or your intranet. You only need to know the URL of the Oracle HTTP Server to access Oracle9i.

Oracle HTTP Server

The middle tier contains the Oracle HTTP Server and the iSQL*Plus Server. The iSQL*Plus Server enables communication and authentication between the iSQL*Plus user interface and Oracle9i. Each iSQL*Plus session is uniquely identified, so you can have multiple concurrent sessions open to Oracle9i.

Oracle9i

Oracle Net components provide communication between the iSQL*Plus Server and Oracle9i in the same way as for a client server installation of Oracle9i.

Configuring iSQL*Plus

This chapter provides information to help you configure the components of iSQL*Plus.

Specific topics discussed are:

- System Requirements
- Installation
- Configuring the Client Tier
- Configuring the Middle Tier
- Configuring the Database Tier
- iSQL*Plus Security
- Configuring Globalization Support

System Requirements

The following subsections describe the system prerequisites for *iSQL*Plus*.

Web Browser (client tier)

You can use any machine with a browser that can connect to the Oracle HTTP Server. The following web browsers are supported:

- Netscape Navigator 4.7 or later.
- Microsoft Internet Explorer 5.0 or later.

Your browser must be set to enable the use of JavaScript and cookies.

While many web browsers and browser versions will work, the functionality and display may be different or restricted. The browser must support the HTML 4.0 Transitional recommendation of the World Wide Web Consortium. See <http://www.w3.org/TR/html4/loose.dtd> for details about HTML declarations.

Users of *iSQL*Plus* may construct arbitrary output text which may include HTML tags. The version of HTML required to run this arbitrary output is your responsibility and may differ from HTML 4.0 Transitional.

Oracle HTTP Server (middle tier)

*iSQL*Plus* is supported when installed with the release of the Oracle HTTP Server included with Oracle9*i* release 9.0.1. *iSQL*Plus* is supported on Microsoft Windows 2000 and Microsoft Windows NT 4.0.

Oracle9*i* (database tier)

*iSQL*Plus* is supported when connected to Oracle9*i*, either installed on the middle tier, or via Oracle Net to a remote Oracle9*i* installation.

Installation

The *iSQL*Plus* Server is installed on the middle tier as a component of Oracle9*i*. During installation on the middle tier, you must select the following components for the *iSQL*Plus* Server to be installed:

- Oracle HTTP Server
- Oracle Net Services
- *SQL*Plus*

Configuring the Client Tier

This section discusses options for configuring your web browser to use *iSQL*Plus*. Specific topics discussed are:

- Adding MIME Types
- Adding Proxy Server Exceptions

Note that your network and browser configuration may not require you to change any settings in your web browser.

Adding MIME Types

Some web browsers may require you to either remove a MIME type definition or application association for files with a .SQL extension, or to create a MIME type or application association for files with a .SQL extension in order to load scripts into *iSQL*Plus*.

For example, to set up an application extension for files with a SQL extension in Netscape Navigator 4.7 for Windows NT:

1. Select Preferences from the Edit menu.
2. Select Applications from the Navigator menu tree.
3. Click the New Type button. On the displayed form, enter

Description of type: SQL files

File extension: SQL

MIME type: text/plain

Application to use: notepad.exe

Uncheck the "use this MIME as the outgoing default for this extension"

If this is not set up in your web browser, you may get an error when you try to load scripts that *iSQL*Plus* cannot identify as text files.

Adding Proxy Server Exceptions

Some configurations of proxy servers may affect the ability of the *iSQL*Plus* user interface to connect with the *iSQL*Plus* Server. If you cannot connect with the *iSQL*Plus* Server, a browser alert "Document contains no data" is displayed when you try to load the *iSQL*Plus* Log In screen. If this situation occurs, you should reconfigure your proxy server or create a proxy exception in your browser for the Oracle HTTP Server running *iSQL*Plus*.

To configure the proxy exceptions setting in Netscape Navigator 4.7 for Windows

1. Select Preferences from the Edit menu.
2. Select Proxies from the Advanced category.
3. Select the Manual proxy configuration radio button.
4. Click View. The Manual proxy configuration dialog is displayed.
5. Enter the Oracle HTTP Server domain for which you do not want to use a proxy in the Exceptions pane.

To configure the proxy exceptions setting in Microsoft Internet Explorer 5.0

1. Select Internet Options from the Tools menu.
2. Click Lan Settings in the Lan Settings pane on the Connections tab.
3. Click Advanced in the Proxy Server pane. This is only available if Use a proxy server is selected.
4. Enter "*" followed by the Oracle HTTP Server domain for which you do not want to use a proxy in the Exceptions pane. For example, to enter an exception for the Oracle HTTP Server domain, *host.domain*, you must enter **.host.domain*.

Configuring the Middle Tier

The Oracle HTTP Server and the iSQL*Plus Server are installed on the middle tier during Oracle9i installation. This section discusses configuration options for the middle tier. Specific topics discussed are:

- Configuring the iSQL*Plus Server
- Configuring the Oracle HTTP Server

Configuring the iSQL*Plus Server

The iSQL*Plus Server is installed during Oracle9i installation on the middle tier. You can:

- Define connection identifiers
- Set the iSQL*Plus session timeout behavior
- Change the appearance of iSQL*Plus screens by modifying the cascading style sheet
- Enable/disable the iSQL*Plus Server

Defining a Connection Identifier

You can create new connection identifiers for use in *iSQL*Plus*. The new definitions are made in the *tnsnames.ora* file on the middle tier machine which is running the *iSQL*Plus* Server.

To create a new connection identifier for *iSQL*Plus*

1. Log in to the middle tier machine which is running the *iSQL*Plus* Server as the system administrator.
2. Change directory to %ORACLE_HOME%\network\admin
3. Open *tnsnames.ora* in a text editor.
4. Add a new line to the end of file in the form:

```
connection identifier = (DESCRIPTION=(ADDRESS_
LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=www.oracle.com)
(PORT=1521)))(CONNECT_DATA=(SID=orashop)))
```

See the *Oracle Net Services Administrator's Guide* for more information about defining connection identifiers.

Setting the *iSQL*Plus* Session TimeOut

Timing out *iSQL*Plus* sessions frees up resources for other users. The timeout settings determine whether sessions will timeout, and how long before they timeout. By default, an *iSQL*Plus* session times out after 60 minutes. You can edit the *iSQL*Plus* configuration file, *isqlplus.conf*, to change the lines affecting the timeout behavior. The syntax of the two lines to change in the configuration file is:

```
iSQLPlusTimeOut {On|Off}
iSQLPlusTimeOutInterval {n}
```

Where *n* is the number of whole minutes of inactivity until the session times out. The default is 60.

When a user tries to use a timed out *iSQL*Plus* session, the Log In screen is displayed and the user is prompted to log in again. The *iSQL*Plus* timeout feature is independent of the Oracle HTTP Server keep alive settings.

Changing the Default Cascading Style Sheet

*iSQL*Plus* uses a cascading style sheet to control the format of the user interface. You can replace the default style sheet with your own style sheet. It is recommended that you do not delete the default style sheet, but rename it so that you can revert to it if necessary.

To replace the default cascading style sheet

1. Navigate to the `%ORACLE_HOME%\sqlplus\admin\iplus\` directory.
2. Rename the existing default style sheet from *iplus.css* to another name.
3. Copy your new style sheet to the `%ORACLE_HOME%\sqlplus\admin\iplus\` directory.
4. Rename your new style sheet to *iplus.css*.
5. You may need to refresh your browser display to see the effects of your new style sheet.

Enabling or Disabling iSQL*Plus

You can edit the Oracle HTTP Server configuration file to disable iSQL*Plus.

To disable the Oracle HTTP Server

1. Stop the Oracle HTTP Server.
2. Change directory to the Oracle HTTP Server configuration directory by entering:

```
cd %ORACLE_HOME%\Apache\Apache\conf
```

3. Open the *oracle_apache.conf* configuration file.
4. Comment out the *isqlplus.conf* include line by inserting a # at the beginning of the line as follows:

```
# include "ORACLE_HOME\sqlplus\admin\isqlplus.conf"
```

Where *ORACLE_HOME* is the hard-coded path of your Oracle home directory.

5. Save your *oracle_apache.conf* file.
6. Check your edits by parsing the edited configuration file. See "Testing the Oracle HTTP Server Configuration File"
7. When you next start the Oracle HTTP Server, iSQL*Plus is disabled.

To re-enable iSQL*Plus, reverse the edit to uncomment the include line in *oracle_apache.conf*.

Configuring the Oracle HTTP Server

Additional iSQL*Plus configuration information must be included in the Oracle HTTP Server configuration file, *httpd.conf*, for the iSQL*Plus Server.

There are two levels of include:

- *httpd.conf* includes the Oracle9i configuration file, *oracle_apache.conf*
- *oracle_apache.conf* includes the iSQL*Plus configuration file, *isqlplus.conf*.

Changes are usually only made to:

- *oracle_apache.conf* to enable Oracle HTTP Server authentication at the User privilege level
- *isqlplus.conf* to disable iSQL*Plus

After making changes to *isqlplus.conf*, the top level Oracle HTTP Server configuration file, *httpd.conf*, should be checked, and then the Oracle HTTP Server stopped and started to implement the configuration file changes.

Testing the Oracle HTTP Server Configuration File

To check the Oracle HTTP Server configuration file, *httpd.conf*, and any included configuration files for errors

1. Open a Windows Command Prompt.
2. Change directory to the Oracle HTTP Server home directory by entering:

```
cd %ORACLE_HOME%\Apache\Apache
```

3. Parse the Oracle HTTP Server *httpd.conf* configuration file by entering:

```
apache -t
```

Any errors in the configuration file are displayed. If there are any errors, edit the included configuration files, *oracle_apache.conf* or *isqlplus.conf*, again to correct them and then test again. If there are no errors, the message "Syntax OK" is displayed.

Starting and Stopping the Oracle HTTP Server

For changes to the Oracle HTTP Server configuration file, *httpd.conf*, and any included configuration files to take affect, you must stop and restart the Oracle HTTP Server if it is running. There is no convenient way to know how many users are currently accessing the server, so it is important to have the server down for the

shortest time. When making changes to configuration files, use the `apache -t` command to parse the `httpd.conf` configuration file and report any errors before starting and stopping the Oracle HTTP Server.

To stop and start the Oracle HTTP Server

1. Open a Windows Command Prompt.
2. Change directory to the Oracle HTTP Server home directory by entering:

```
cd %ORACLE_HOME%\Apache\Apache\
```

3. Stop the running Oracle HTTP Server by entering:

```
apache -k shutdown
```

4. Start the Oracle HTTP Server by entering

```
apache -k start
```

Configuring the Database Tier

Oracle9i is installed on the database tier. The database tier may be physically separate from the middle tier and accessed using Oracle Net, or it may be the same physical machine as used by the middle tier. For further information about configuring Oracle9i, see the Oracle9i documentation.

iSQL*Plus Security

Each iSQL*Plus log in is uniquely identified, so you can:

- Connect multiple times from the same machine
- Connect multiple times from different machines

iSQL*Plus supports this stateful behavior by storing session context information in the Oracle HTTP Server. You must ensure that your listener always routes HTTP requests to the same server, otherwise the session context will not be found.

However, you may find it useful to start more than one Oracle HTTP Server to distribute user load across the multiple servers.

There are two main areas to consider for security and user authentication when using iSQL*Plus:

- The HTTP protocol connection between the web browser and the Oracle HTTP Server.

- The Oracle Net connection between the iSQL*Plus module (in the Oracle HTTP Server) and Oracle9i.

In this release of iSQL*Plus, security for the connection between the web browser and the Oracle HTTP Server is provided by standard HTTPS, which is fully supported by Oracle. It enables secure listener connections with an Oracle-provided encryption mechanism via the Secure Sockets Layer (SSL). It can be implemented when installing the Oracle HTTP Server by installing the *mod_ssl* module. For detailed information about implementing HTTPS security in Oracle, see the *Oracle Advanced Security Administrator's Guide*.

The Oracle Net connection between the iSQL*Plus module and Oracle9i provides the same security as in previous client server architectures. For more information about Oracle Net connection security, see the *Oracle Net Services Administrator's Guide* and the *Oracle Advanced Security Administrator's Guide*.

Using Administration Privileges

There are two modes of access to iSQL*Plus:

- Connect as a normal User, the default mode.
- Connect with AS SYSDBA or AS SYSOPER privileges.

When you log in with User privileges, you cannot use the SQL*Plus CONNECT command to reconnect with AS SYSDBA or AS SYSOPER privileges, and therefore cannot perform privileged operations such as shutting down the server. Any attempt to connect with AS SYSDBA or AS SYSOPER privileges from a user session will fail with the error message "SP2-0563: Insufficient privileges".

Enabling User Security

You may want to limit the users who can access iSQL*Plus. Oracle HTTP Server authentication is required for AS SYSDBA and AS SYSOPER connections, but not for User connections. You can edit the *isqlplus.conf* file to enable Oracle HTTP Server authentication for User connections by changing the following lines:

```
<Location /isqlplus>
    SetHandler iplus-handler
    Order deny,allow
    Allow from all
</Location>
```

to:

```
<Location /isqlplus>
  SetHandler iplus-handler
  Order deny,allow
  AuthType Basic
  AuthName 'iSQL*Plus'
  AuthUserFile %ORACLE_HOME%\sqlplus\admin\iplus.pw
  Require valid-user
</Location>
```

In this case, *iplus.pw* is suggested as the file to contain the Oracle HTTP Server authentication usernames and passwords for User connections. Now, whenever a User connection is requested, users are not only required to enter their Oracle9i username and password, but they are also prompted to enter an Oracle HTTP Server authentication username and password.

Adding Entries to an Oracle HTTP Server Authentication File

To connect with SYSDBA or SYSOPER privileges, your username and password must be added to the iSQL*Plus authentication file for the Oracle HTTP Server. On installation, the authentication file is created with no user entries at %ORACLE_HOME%\sqlplus\admin\iplusdba.pw. The username and password used in the authentication file is independent of the Oracle9i username and password.

If you have enabled Oracle HTTP Server authentication for User connections, you need to create a separate authentication file to contain username/password entries for User level connections. See "Enabling User Security" for information about enabling User level Oracle HTTP Server authentication.

To create a new user entry in an Oracle HTTP Server authentication file.

1. Log in to the Oracle HTTP Server as the Oracle HTTP Server administrator.
2. Run the *htpasswd* utility to add users to the authentication file. For AS SYSDBA or AS SYSOPER users, use the form:

```
htpasswd %ORACLE_HOME%\sqlplus\admin\iplusdba.pw username
```

For User connections, where *iplus.pw* has been created as an empty authentication file, use the form:

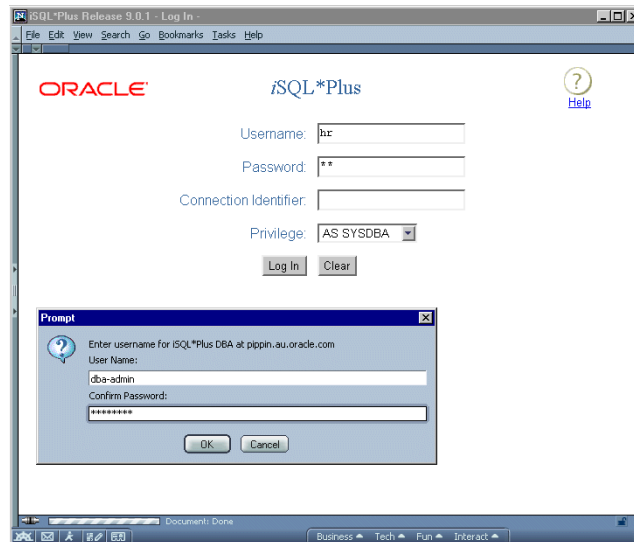
```
htpasswd %ORACLE_HOME%\sqlplus\admin\iplus.pw username
```

In both cases you are prompted for the associated password. For further information about *htpasswd*, see the Oracle HTTP Server documentation.

Logging In Using Oracle HTTP Server Authentication

To connect to a database instance with SYSDBA or SYSOPER privileges, or as a User with Oracle HTTP Server authentication enabled:

1. Open a browser window and start a new iSQL*Plus session.
2. Enter your Oracle9i username and password.
3. Enter a Connection Identifier for the database connection you want, or leave the Connection Identifier blank to connect to the default database.
4. Select the AS SYSDBA, AS SYSOPER, or User privileges you want for this database connection from the Privileges: dropdown list on the Log In screen.
5. Click Log In. You are prompted to enter your Oracle HTTP Server authentication username and password. The username and password must be a valid entry in the Oracle HTTP Server authentication file. A login screen similar to the following is displayed:



6. Click OK. You are connected to the selected database with the SYSDBA, SYSOPER, or User privileges you requested.

Security Usage Notes

The following notes may assist you in understanding and configuring iSQL*Plus:

- If the user who starts the Oracle HTTP Server is not a member of the ORA_DBA group, then the Operating System (OS) authentication that allows privileged "/ as sysdba" connections will not work. However, connections using the Privileges: field on the Log In screen to enable SYSDBA or SYSOPER privileged connections using the Oracle HTTP Server authentication file will work.
- Once you have successfully logged in with AS SYSDBA or AS SYSOPER privileges and authenticated with your Oracle HTTP Server authentication username and password, you may not be required to re-authenticate to the Oracle HTTP Server until you restart your browser. However, you are still required to log in with your Oracle9i username and password.
- The Product User Profile (PUP) tables are per user, per database as for client server installations. See Appendix B, "Security" for more information about PUP tables.
- The global configuration file *glogin.sql* is read from the middle tier machine as for a client server installation. No user *login.sql* file is read.

Configuring Globalization Support

To display another language in iSQL*Plus, you must configure all three tiers. The following example is for Japanese using the Unicode (UTF-8) character set.

To use the UTF-8 character encoding for Japanese:

Web Browser - client tier:

1. Ensure you have a font available that supports the Unicode character set.

Microsoft Language Packs and updates are available for Windows operating systems to provide international language support at:

<http://windowsupdate.microsoft.com/R712/v31site/x86/w98/en/thanksstart.htm>

or you can find information about Asian language support for Windows at:

<http://www.microsoft.com/windows/ie/features/ime.asp>

Netscape information about fonts, and font downloads to support international language sets can be found at:

<http://home.netscape.com/eng/intl/basics.html>.

2. Change the character encoding in your web browser:

For Microsoft Internet Explorer:

- a. Select Encoding from the View menu.
- b. Select Unicode (UTF-8).

For Netscape Navigator:

- a. Select Character Set from the View menu.
- b. Click Unicode (UTF-8).

3. Set the fonts associated with the UTF-8 character encoding:

For Microsoft Internet Explorer:

- a. Select Internet Options... from the Tools menu.
- b. Select the General tab in the Internet Options dialog box and click Fonts...
- c. Choose a language script and the web page (proportional) font and plain text (fixed-width) font to be used with it.
- d. You can install fonts and support for additional languages by selecting Windows Update on the Tools menu

For Netscape Navigator:

- a. Select Preferences from the Edit menu.
- b. Select Fonts under Appearance.
- c. Choose Unicode from the Encoding dropdown list.
- d. Select the variable and fixed pitch fonts that support the Unicode character set from the Variable and Fixed Pitch dropdown lists.

Oracle HTTP Server - middle tier:

Set the Globalization Support variables you want in the Oracle HTTP Server operating system. Every iSQL*Plus session inherits this environment.

1. In Windows, open System from the Control Panel.
2. Create a new environment variable, NLS_LANG, with a value of Japanese_Japan.UTF8.
3. Restart the Oracle HTTP Server.

Oracle9i - database tier:

Create your Oracle9i database with the UTF8 character set.

Note: Some *iSQL*Plus* error messages are in English in this release. See your *Oracle9i Globalization Support Guide* for more information about Configuring Globalization Support.

The *i*SQL*Plus User Interface

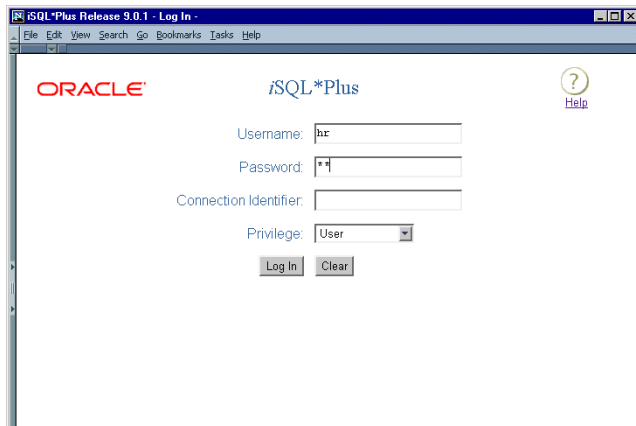
This chapter describes the *i*SQL*Plus user interface and options.

Specific topics discussed are:

- The *i*SQL*Plus Log In Screen
- The *i*SQL*Plus Work Screen
- The Change Password Screen

The iSQL*Plus Log In Screen

After you have successfully connected to iSQL*Plus, the Log In screen is displayed:



Different web browsers, and the size of the web browser window, may affect the appearance and layout of the Log In screen.

? Help icon Click the Help icon to access the *iSQL*Plus User's Guide and Reference* in a separate browser window.

Username: Enter a valid username to connect to Oracle9i. You can optionally enter your password separated by a / after the username.

Password: Enter a valid password for the username. Your password is not displayed.

Connection Identifier: Leave this field blank to use the default Oracle database if one exists, otherwise enter a connection identifier to specify a remote database you want to connect to. You can use either a TNS alias, or the full connect identifier, for example:

```
(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=www.oracle.com)
(PORT=1521))))(CONNECT_DATA=(SID=orashop)))
```

If you use a TNS alias, it must be specified on the machine running the iSQL*Plus Server, which may not be the same machine from which you run your web browser.

Privilege: The Privilege dropdown list has three options:

- **User**—is the default connection. iSQL*Plus connects to the specified database with no administrator privileges.
- **AS SYSDBA**—connects to the specified database with SYSDBA privileges.
- **AS SYSOPER**—connects to the specified database with SYSOPER privileges.

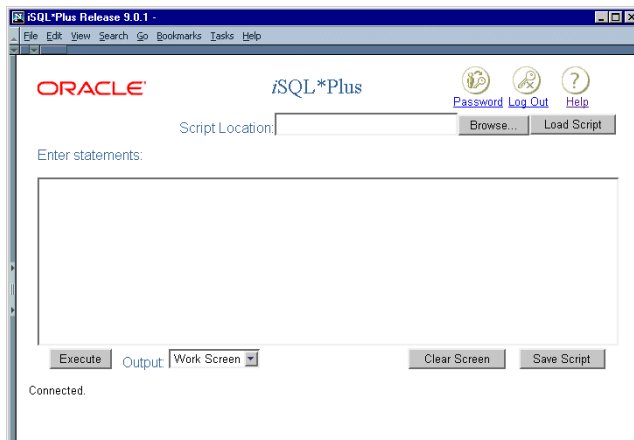
To connect with either SYSDBA or SYSOPER privileges, your username and password must be added to the Oracle HTTP Server authentication file. For further information about connection privileges, see "Using Administration Privileges" in Chapter 2.

Log In Click the Log In button to log in to iSQL*Plus with the supplied username, password, connection identifier and privilege.


Clear Click the Clear button to clear the fields on the Log In screen and allow you to re-enter login details.


The iSQL*Plus Work Screen


After successfully logging in, the iSQL*Plus Work screen is displayed.



Different web browsers, and the size of the web browser window, may affect the appearance and layout of the Work screen.

 **Password Icon** Click the Password icon to change your password. The Change Password screen is displayed.

 **Log Out Icon** Click the Log Out icon to exit iSQL*Plus. You are returned to the Log In screen.

 **Help Icon** Click the Help icon to access the *iSQL*Plus User's Guide and Reference* in a separate browser window.

Enter statements: Enter SQL statements, PL/SQL blocks, or iSQL*Plus commands. This area is also referred to as the *input area*.

Script location: Enter the path and name of a script that you want to load for editing or execution.

Browse... Click the Browse... button to search for a script file that you want to load for editing or execution. When you select the file, its path and name are entered in the Script location: field.

Load Script Click the Load Script button to load the script specified in the Script location: field into the iSQL*Plus input area for editing or execution.

Execute: Click the Execute button to execute the contents of the input area. Depending on the Output dropdown list selection, the results of the execution are displayed, or saved to a file.

Output: The Output dropdown list has three options:

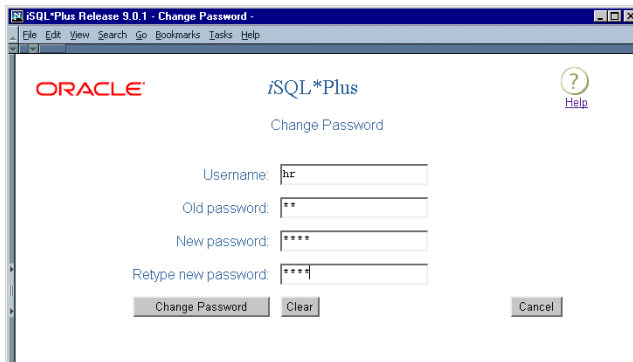
- **Work Screen**—when the contents of the input area are executed, the resulting output is displayed on screen under the input area. This is the default.
- **File**—when the contents of the input area are executed, the resulting output is saved to a file. You are prompted to enter the name of the file. As the output is in HTML format, it is useful to give the saved output file a .HTM or .HTML extension.
- **Window**—when the contents of the input area are executed, the resulting output is displayed in a new browser window.


Clear Screen Click the Clear Screen button to clear all statements in the input area, and all displayed output. It does not clear the SQL buffer, nor does it clear any variable values altered by options of the SET command.

Save Script Click the Save Script button to save the contents of the input area to a file. You are prompted to enter the name of the file. The file extension you choose is for your convenience. It may be convenient to identify scripts with an extension of .SQL.

The Change Password Screen

You can change your password in the Change Password screen. If you have database administrator privileges, you can change the password of other users. Click the Change Password icon to display the Change Password screen.



 **Help icon** Click the Help icon to access the *iSQL*Plus User's Guide and Reference* in a separate browser window.

Username: Enter the username.

Old password: Enter the current password for the username.

New password: Enter the new password.

Retype new password: Enter the new password again.

Change Password Click the Change Password button to change the password for the username you entered.

Clear Click the Clear button to clear all fields.

Cancel Click the Cancel button to return to the *iSQL*Plus Work* screen without changing the password.

The Change Password screen is automatically displayed if you log into *iSQL*Plus* and your password has expired.

Using *iSQL*Plus*

This chapter explains how to run and use *iSQL*Plus*.

Specific topics discussed are:

- Running *iSQL*Plus*
- Entering Statements
- Loading Scripts
- Entering Substitution Variables
- Creating Dynamic Reports
- Using SET and COLUMN Commands
- Getting Help
- Exiting *iSQL*Plus*

Running iSQL*Plus

The Oracle HTTP Server and the iSQL*Plus Server must be running on the middle tier before you can run iSQL*Plus.

To run iSQL*Plus

1. Enter the Uniform Resource Identifier (URI) of iSQL*Plus in the Location field of your web browser, for example:

```
http://host.domain/isqlplus
```

where *host.domain* is the URI for the Oracle HTTP Server you want to use. The iSQL*Plus Log In screen is displayed.

Each successful login is uniquely identified, so you can have multiple iSQL*Plus sessions running from the same machine, or from multiple client machines.

2. Enter your Username, Password and Connection Identifier.
3. Select the Privilege level you want to use to connect to Oracle9i.
4. Click the Log In button. The iSQL*Plus Work screen is displayed.

When you connect with AS SYSDBA or AS SYSOPER privileges, the URI changes from the form

```
http://host.domain/isqlplus
```

to the form

```
http://host.domain/isqlplusdba
```

When you are connected through the isqlplusdba URI, the Oracle HTTP Server authentication permits AS SYSDBA or AS SYSOPER connections through the Log In screen, or through a CONNECT command, but the Oracle9i username and password authentication may still prevent access.

Entering Statements

You enter multiple SQL statements, PL/SQL blocks, and SQL*Plus commands in the input area. You can use backspace and delete keys, and you can cut and paste using your browser's edit keys to edit the statements in the input area. You can also cut or copy scripts or statements from other applications such as Notepad, and paste them directly into the input field.

You can save scripts using the Save Script button. *iSQL*Plus* saves scripts to a text file. You can load scripts with the Load Script button. Saving and loading scripts may be useful when editing and testing scripts.

Click the Execute button to execute the contents of the input area. The results of your query are displayed below the input area. The last SQL statement or PL/SQL block entered is stored in a buffer in the *iSQL*Plus* Server. You can see this statement with the LIST command and you can re-execute it by entering a / in the empty input area, and clicking the Execute button.

*iSQL*Plus* executes a SQL or PL/SQL statement at the end of the input area, even if it is incomplete or does not have a final ";" or "/". If you intend to run *iSQL*Plus* scripts in the SQL*Plus command line, you should make sure you use a ";" or "/" to terminate your statements.

Remember that *iSQL*Plus* retains the state of your current SET variables and other options from one execution to the next. If you use the back button of your browser to find a script you previously entered in the input area, and re-execute it, you may get different results from those previously obtained, depending on the current option values.

Some SQL*Plus commands have no logical sense or are not applicable in *iSQL*Plus*. See Appendix C, "Unsupported SQL*Plus Commands" for a list of unsupported SQL*Plus commands.

Loading Scripts

Editing complex scripts in another text editor with more sophisticated tools for search and replace, and other text manipulation, may be an advantageous way to prepare original scripts or variants of existing scripts which can be subsequently loaded into *iSQL*Plus* for editing or execution.

You can load any SQL script into *iSQL*Plus* that you can access from your workstation. Make sure that scripts you load are valid SQL scripts for use with *iSQL*Plus*.

Some web browsers may require you to create a MIME type or application association for files with a .SQL extension in order to load them into *iSQL*Plus*. See "Adding MIME Types" in Chapter 2 for information on how to create a MIME or application association.

Entering Substitution Variables

System variables specified with the SET command can affect *iSQL*Plus* behavior. SET DEFINE, SET ESCAPE, SET VERIFY ON, and SET CONCAT affect variable substitution behavior, and should be set in *iSQL*Plus* before attempting to execute a script. For further information about these SET options, see the SET command in Chapter 5, "Command Reference".

'&' variables are parsed in *iSQL*Plus* before the script is sent to the SQL*Plus engine, and then in the SQL*Plus engine. You should synchronize variable substitution with one of the two following options:

- Enter SET DEFINE ON to set *iSQL*Plus* to always prompt for substitution variables. Click the Execute button to execute the command.

Enter your script using '&' and '&&' as the prefix for variables. Do not use DEFINE, or UNDEFINE. Click the Execute button to execute the script. *iSQL*Plus* prompts you for values for all the substitution variables in your script. At the end of script execution, any double ampersand values contained in the script remain DEFINED. This means that you will not be prompted to enter values for these variables again, until they have been UNDEFINED, or you log out of *iSQL*Plus*. If this is not the behavior you want, then use a single ampersand as the prefix for variables in your script. You are always prompted for value substitution for variables created with a single ampersand prefix.

- Enter SET DEFINE OFF to set *iSQL*Plus* never to prompt for substitution variables. Click the Execute button to execute the command.

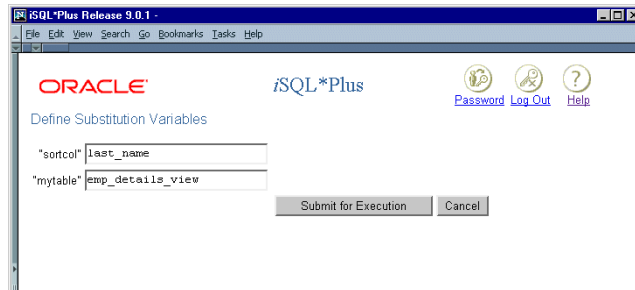
Enter your script. Enter SET DEFINE ON, and DEFINE all variables with '&' or '&&' before they are referenced. You can also use COLUMN ... *new_value* or COLUMN ... *old_value* to instantiate substitution variables. Click the Execute button to execute the script. *iSQL*Plus* does not prompt you for values for substitution variables. In this usage, be aware that *iSQL*Plus* substitutes a NULL value for any variable you do not DEFINE in the script.

The Define Substitution Variables Screen


When *iSQL*Plus* executes a script containing substitution variables, the Define Substitution Variables screen is displayed. For example, when you enter:


```
BREAK ON &SORTCOL
SELECT &SORTCOL, SALARY
FROM &MYTABLE
WHERE SALARY > 12000
ORDER BY &SORTCOL
```


iSQL*Plus displays:



Different web browsers, and the size of the web browser window, may affect the appearance and layout of the Define Substitution Variables screen.

 **Password Icon** Click the Password icon to change your password. The Change Password screen is displayed.

 **Log Out Icon** Click the Log Out icon to exit iSQL*Plus. You are returned to the Log In screen.

 **Help Icon** Click the Help icon to access the *iSQL*Plus User's Guide and Reference* in a separate browser window.

Define Substitution Variables Enter values for the substitution variables. For example, enter LAST_NAME in the *sortcol* field, and EMP_DETAILS_VIEW in the *mytable* field of the example script.

Submit for Execution Click the Submit for Execution button to execute the script in the input area with the substitution variable values you entered. Depending on the Output dropdown list selection, the results of the execution are displayed, or saved to a file.

Cancel Click the Cancel button to return to the Work screen without executing the script in the input area.

Creating Dynamic Reports

You can create dynamic reports, and pass variables to scripts by sending *iSQL*Plus* a request to run a script from a Uniform Resource Identifier (URI). The script must be available through HTTP. *iSQL*Plus* executes the script, using any HTML form field values as parameters, and returns the results in a new browser window.

You will not be prompted for undefined variables. You should take care that there are no variables that have not been DEFINED in your script, or explicitly passed as parameters.

You can also include username and password information in the request. You should carefully consider the security implications of including usernames and passwords in HTML files. If you do not include a username or password, *iSQL*Plus* prompts you to enter login information when you run the script.

Example 4–1 Creating a Dynamic Report

Create and save the following script to a file called *script.sql* on your Oracle HTTP Server.

```
SET PAGESIZE 200
SELECT *
FROM EMP_DETAILS_VIEW
ORDER BY LAST_NAME, EMPLOYEE_ID
/
```

Create an HTML file which contains:

```
<HTML>
<HEAD>
<TITLE>iSQL*Plus Dynamic Report</TITLE>
</HEAD>
<BODY>
<H1>iSQL*Plus Report</H1>
<A HREF="http://host.domain/isqlplus?script=http://host.domain/script.sql">
Run Employee Report</A>
</BODY>
</HTML>
```

Replace *host.domain* with the host and domain names of your Oracle HTTP Server. Save the HTML file on your Oracle HTTP Server.

Load the HTML file in your web browser and click on "Run Employee Report". *iSQL*Plus* requests your username and password. Log in to *iSQL*Plus*. *iSQL*Plus* executes the script and displays the results in your web browser.

Example 4-2 Creating a Dynamic Report with Parameters

Create and save the following script to a file called *employee_name.sql* on your Oracle HTTP Server.

```
SET VERIFY OFF
SET PAGESIZE 200
SET FEEDBACK OFF
SET MARKUP HTML ENTMAP OFF
PROMPT <H1>Employee Details for Employee(s) with Last Name like &last_name%</H1>
SET MARKUP HTML ENTMAP ON
SELECT *
FROM EMPLOYEES
WHERE UPPER(last_name) LIKE UPPER('&last_name%')
/
```

Create an HTML file which contains:

```
<HTML>
<HEAD>
<TITLE>iSQL*Plus Dynamic Report</TITLE>
</HEAD>
<BODY>
<H1>iSQL*Plus Report</H1>
<H2>Query by Last Name</H2>
<FORM METHOD=get ACTION="http://host.domain/isqlplus">
<INPUT TYPE="hidden" NAME="script" VALUE="http://host.domain/employee_name.sql">
Enter last name of employee: <INPUT TYPE="text" NAME="last_name" SIZE="20">
<INPUT TYPE="submit" VALUE="Run Report">
</FORM>
</BODY>
</HTML>
```

The name of the INPUT TYPE should be the same as either a column or substitution variable in your script, for example

```
<INPUT TYPE="text" NAME="last_name" SIZE="20">
```

maps to the substitution variable *&last_name* in the *employee_name.sql* script.

Replace *host.domain* with the host and domain names of your Oracle HTTP Server. Save the HTML file on your Oracle HTTP Server.

Load the HTML file in your web browser. Enter a name or partial name in the text field, for example, "Fay". Click the Run Report button. iSQL*Plus executes the script and displays the results in your web browser.

Example 4–3 Creating a Dynamic Script with Parameters and Login Details

Create and save the following script to a file called *employee_id.sql* on your Oracle HTTP Server.

```
SET VERIFY OFF
SET PAGESIZE 200
SET MARKUP HTML ENTMAP OFF
PROMPT <H1>Employee Details for Employee Number &eid</H1>
SET MARKUP HTML ENTMAP ON
SELECT *
FROM EMPLOYEES
WHERE EMPLOYEE_ID = &eid
/
```

Create an HTML file which contains:

```
<HTML>
<HEAD>
<TITLE>iSQL*Plus Dynamic Report</TITLE>
</HEAD>
<BODY>
<H1>iSQL*Plus Report</H1>
<H2>Query by Employee ID</H2>
<FORM METHOD=get ACTION="http://host.domain/isqlplus">
<INPUT TYPE="hidden" NAME="userid" VALUE="hr/your_secret_password">
<INPUT TYPE="hidden" NAME="script" VALUE="http://host.domain/employee_id.sql">
Enter employee identification number: <INPUT TYPE="text" NAME="eid" SIZE="10">
<INPUT TYPE="submit" VALUE="Run Report">
</FORM>
</BODY>
</HTML>
```

Replace *host.domain* with the host and domain names of your Oracle HTTP Server, and *hr/your_secret_password* with a valid userid and password. Save the HTML file on your Oracle HTTP Server.

Load the HTML file in your web browser. Enter an employee identification number in the text field, for example, "105". Click the Run Report button. iSQL*Plus executes the script and displays the results in your web browser.

Using SET and COLUMN Commands

iSQL*Plus remembers the state of your current SET command variables from one execution to the next. If you use the Back button on your web browser and then re-execute scripts, iSQL*Plus may display the results differently, according to any commands executed since the original script was executed. For example, a new SET PAGESIZE command may set column headings to appear in different positions.

The COLUMN command, and the SET LINESIZE and SET WRAP commands have an effect on the buffer sizes used by iSQL*Plus. If words in your query output include unexpected white spaces or are truncated, make sure the column size is large enough to accommodate the word, and that word wrapping is set to ON.

Getting Help

The online *iSQL*Plus User's Guide and Reference* provides help and syntax specific to iSQL*Plus. Click the Help button to access the *iSQL*Plus User's Guide and Reference*. The *iSQL*Plus User's Guide and Reference* is displayed in a new browser window.

Exiting iSQL*Plus

To exit iSQL*Plus, click the Log Out button. It is recommended that you always use the Log Out button to exit iSQL*Plus to free up system and server resources.

Command Reference

This chapter contains descriptions of SQL*Plus commands that are accessible through *iSQL*Plus*. Use this chapter for reference only. Command descriptions are listed alphabetically, and each description contains the following parts:

Syntax	Shows how to enter the command and provides a brief description of the basic uses of the command. Refer to "Conventions in Code Examples" in the Preface for an explanation of the syntax notation
Terms	Describes the function of each term or clause appearing in the syntax.
Usage	Provides additional information on uses of the command and on how the command works.
Examples	Gives one or more examples of the command.

A summary table that lists and briefly describes SQL*Plus commands precedes the individual command descriptions.

SQL*Plus Command Summary

Command	Page	Description
@	5-5	Runs the SQL*Plus statements in the specified command file. The command file can be called from the local file system or from a web server.
@@	5-7	Runs a command file. This command is similar to the @ (“at” sign) command. It is useful for running nested command files because it looks for the specified command file in the same path as the command file from which it was called.
/	5-9	Executes the SQL command or PL/SQL block.
ARCHIVE LOG	5-10	Starts or stops the automatic archiving of online redo log files, manually (explicitly) archives specified redo log files, or displays information about redo log files.
ATTRIBUTE	5-13	Specifies display characteristics for a given attribute of an Object Type column, and lists the current display characteristics for a single attribute or all attributes.
BREAK	5-15	Specifies where and how formatting will change in a report, or lists the current break definition.
BTITLE	5-20	Places and formats a specified title at the bottom of each report page, or lists the current BTITLE definition.
CLEAR	5-21	Resets or erases the current clause or setting for the specified option, such as BREAKS or COLUMNS.
COLUMN	5-23	Specifies display characteristics for a given column, or lists the current display characteristics for a single column or for all columns.
COMPUTE	5-34	Calculates and prints summary lines, using various standard computations, on subsets of selected rows, or lists all COMPUTE definitions.
CONNECT	5-40	Connects a given user to Oracle.
COPY	5-42	Copies results from a query to a table in a local or remote database.
DEFINE	5-46	Specifies a user variable and assigns it a CHAR value, or lists the value and variable type of a single variable or all variables.
DESCRIBE	5-48	Lists the column definitions for the specified table, view, or synonym or the specifications for the specified function or procedure.

Command	Page	Description
DISCONNECT	5-54	Commits pending changes to the database and logs the current user off Oracle, but does not exit SQL*Plus.
EXECUTE	5-55	Executes a single PL/SQL statement.
HELP	5-56	Accesses the SQL*Plus help system.
LIST	5-57	Lists one or more lines of the buffer.
PRINT	5-59	Displays the current value of a bind variable.
PROMPT	5-61	Sends the specified message to the user's screen.
RECOVER	5-62	Performs media recovery on one or more tablespaces, one or more datafiles, or the entire database.
REMARK	5-69	Begins a comment in a command file.
REPFOOTER	5-70	Places and formats a specified report footer at the bottom of each report, or lists the current REPFOOTER definition.
REPHEADER	5-72	Places and formats a specified report header at the top of each report, or lists the current REPHEADER definition.
RUN	5-75	Lists and runs the SQL command or PL/SQL block currently stored in the buffer.
SET	5-76	Sets a system variable to alter the SQL*Plus environment for your current session.
SHOW	5-96	Shows the value of a SQL*Plus system variable or the current SQL*Plus environment.
SHUTDOWN	5-101	Shuts down a currently running Oracle instance.
START	5-103	Runs the SQL*Plus statements in the specified command file. The command file can only be called from a <i>uri</i> .
STARTUP	5-105	Starts an Oracle instance and optionally mounts and opens a database.
TIMING	5-108	Records timing data for an elapsed period of time, lists the current timer's title and timing data, or lists the number of active timers.
TTITLE	5-110	Places and formats a specified title at the top of each report page, or lists the current TTITLE definition.
UNDEFINE	5-114	Deletes one or more user variables that you defined either explicitly (with the DEFINE command) or implicitly (with an argument to the START command).

Command	Page	Description
VARIABLE	5-115	Declares a bind variable that can be referenced in PL/SQL.
WHENEVER OSERROR	5-122	Performs the specified action if an operating system command generates an error.
WHENEVER SQLERROR	5-123	Performs the specified action if a SQL command or PL/SQL block generates an error.

@ ("at" sign)

Syntax

`@{uri|file_name[.ext]} [arg...]`

Runs the SQL*Plus statements in the specified command file. The command file can be called from the local file system or from a web server. *uri* is only supported on Windows platforms in this release. Only the *uri* form is supported in *iSQL*Plus*.

Terms

Refer to the following for a description of the term or clause:

uri

Specifies the Uniform Resource Identifier of a script to run on the specified web server. SQL*Plus supports HTTP, FTP and gopher protocols.

file_name[.ext]

Represents the command file you wish to run. If you omit *ext*, SQL*Plus assumes the default command-file extension (normally SQL).

When you enter `@ file_name.ext`, SQL*Plus searches for a file with the filename and extension you specify in the current default directory. If SQL*Plus does not find such a file, SQL*Plus will search a system-dependent path to find the file. Some operating systems may not support the path search. See the Oracle installation and user's manual(s) provided for your operating system for specific information related to your operating system environment.

arg...

Represent data items you wish to pass to parameters in the command file. If you enter one or more arguments, SQL*Plus substitutes the values into the parameters (&1, &2, and so forth) in the command file. The first argument replaces each occurrence of &1, the second replaces each occurrence of &2, and so forth.

The @ command DEFINES the parameters with the values of the arguments; if you run the command file again in this session, you can enter new arguments or omit the arguments to use the current values.

Usage

In a command file, you can include any command you would normally enter interactively (typically, SQL, SQL*Plus commands, or PL/SQL blocks).

An EXIT or QUIT command used in a command file terminates SQL*Plus.

The @ command functions similarly to START.

If the START command is disabled, this will also disable the @ command. See START in this chapter for information on the START command.

SQL*Plus removes the SQLTERMINATOR (a semicolon by default) before the @ command is issued. If you require a semicolon in your command, add a second SQLTERMINATOR. See the SQLTERMINATOR variable of the SET command in this chapter for more information.

Examples



To run a command file named PRINTRPT with the extension SQL, enter

```
@PRINTRPT
```



To run a command file named WKRPT with the extension QRY, enter

```
@WKRPT.QRY
```

You can run a script named YEAREND specified by a Uniform Resource Identifier, and pass values to variables referenced in YEAREND in the usual way:



```
@HTTP://HOST.DOMAIN/YEAREND.SQL VAL1 VAL2  
@FTP://HOST.DOMAIN/YEAREND.SQL VAL1 VAL2  
@GOPHER://HOST.DOMAIN/YEAREND.SQL VAL1 VAL2
```

On a web server configured to serve SQL reports, you could request SQL*Plus to execute a dynamic script by using:



```
@HTTP://HOST.DOMAIN/SCRIPTSERVER?ENDOFYEAR VAL1 VAL2
```

@@ (double “at” sign)

Syntax

@@file_name[.ext]

Runs a command file. This command is identical to the @ (“at” sign) command. It is useful for running nested command files because it has the additional functionality of looking for the specified command file in the same path or *uri* as the command file from which it was called. *uri* is only supported on Windows platforms in this release. Only the *uri* form is supported in *iSQL*Plus*.

Terms

Refer to the following for a description of the term or clause:

file_name[.ext]

Represents the nested command file you wish to run. If you omit *ext*, *SQL*Plus* assumes the default command-file extension (normally *SQL*).

When you enter *@@file_name.ext* from within a command file, *SQL*Plus* runs *file_name.ext* from the same directory as the command file.

When you enter *@@file_name.ext* interactively, *SQL*Plus* runs *file_name.ext* from the current working directory or from the same *uri* as the command file from which it was called. If *SQL*Plus* does not find such a file, *SQL*Plus* searches a system-dependent path to find the file. Some operating systems may not support the path search. See the Oracle installation and user’s manual provided for your operating system for specific information related to your operating system environment.

Usage

You can include in a command file any command you would normally enter interactively (typically, *SQL* or *SQL*Plus* commands).

An *EXIT* or *QUIT* command used in a command file terminates *SQL*Plus*.

The @@ command functions similarly to *START*.

If the *START* command is disabled, this will also disable the @@ command. For more information, see the *START* command later in this chapter.

*SQL*Plus* removes the *SQLTERMINATOR* (a semicolon by default) before the @@ command is issued. A workaround for this is to add another *SQLTERMINATOR*.

See the `SQLTERMINATOR` variable of the `SET` command in this chapter for more information.

Examples



Suppose that you have the following command file named `PRINTRPT`:

```
SELECT DEPARTMENT_ID, CITY FROM EMP_DETAILS_VIEW WHERE SALARY>12000;  
@EMPRPT  
@@ WKRPT
```

When you `START PRINTRPT` and it reaches the `@` command, it looks for the command file named `EMPRPT` in the current working directory and runs it. When `PRINTRPT` reaches the `@@` command, it looks for the command file named `WKRPT` in the same path as `PRINTRPT` and runs it.

Suppose that the same command file `PRINTRPT` was located on a web server and you ran it with `START HTTP://HOST.DOMAIN/PRINTRPT`. When it reaches the `@` command, it looks for the command file named `EMPRPT` in the current local working directory and runs it. When `PRINTRPT` reaches the `@@` command, it looks for the command file named `WKRPT` in the same *uri* as `PRINTRPT` and runs it.

/ (slash)

Syntax

/

Executes the SQL command or PL/SQL block currently stored in the SQL buffer.

Usage

You can enter a slash (/) at the command prompt or at a line number prompt of a multi-line command.

The slash command functions similarly to RUN, but does not list the command in the buffer on your screen.

Executing a SQL command or PL/SQL block using the slash command will not cause the current line number in the SQL buffer to change unless the command in the buffer contains an error. In that case, SQL*Plus changes the current line number to the number of the line containing the error.

Examples



Type the following SQL query:

```
SELECT CITY, COUNTRY_NAME
FROM EMP_DETAILS_VIEW
WHERE SALARY=12000;
```

Enter a slash (/) at the command prompt to re-execute the command in the buffer:



/



CITY	COUNTRY_NAME
-----	-----
Seattle	United States of America
Oxford	United Kingdom
Seattle	United States of America

ARCHIVE LOG

Syntax

```
ARCHIVE LOG {LIST|STOP}[(START|NEXT|ALL|integer) [TO destination]]
```

Starts or stops automatic archiving of online redo log files, manually (explicitly) archives specified redo log files, or displays information about redo log files.

Terms

Refer to the following list for a description of each term or clause:

LIST

Requests a display that shows the range of redo log files to be archived, the current log file group's sequence number, and the current archive destination (specified by either the optional command text or by the initialization parameter LOG_ARCHIVE_DEST).

If you are using both ARCHIVELOG mode and automatic archiving, the display might appear like:

```
ARCHIVE LOG LIST
```

Database log mode	Archive Mode
Automatic archival	Enabled
Archive destination	/vobs/oracle/dbs/arch
Oldest online log sequence	221
Next log sequence to archive	222
Current log sequence	222

Since the log sequence number of the current log group and the next log group to archive are the same, automatic archival has archived all log groups up to the current one.

If you are using ARCHIVELOG but have disabled automatic archiving, the last three lines might look like:

Oldest online log sequence	222
Next log sequence to archive	222
Current log sequence	225

If you are using NOARCHIVELOG mode, the “next log sequence to archive” line is suppressed.

The log sequence increments every time the Log Writer begins to write to another redo log file group; it does not indicate the number of logs being used. Every time an online redo log file group is reused, the contents are assigned a new log sequence number.

STOP

Disables automatic archival. If the instance is still in ARCHIVELOG mode and all redo log file groups fill, database operation is suspended until a redo log file is archived (for example, until you enter the command ARCHIVE LOG NEXT or ARCHIVE LOG ALL).

START

Enables automatic archiving. Starts the background process ARCH, which performs automatic archiving as required. If ARCH is started and a filename is supplied, the filename becomes the new default archive destination. ARCH automatically starts on instance startup if the initialization parameter LOG_ARCHIVE_START is set to TRUE.

NEXT

Manually archives the next online redo log file group that has been filled, but not yet archived.

ALL

Manually archives all filled, but not yet archived, online redo log file groups.

integer

Causes archival of the online redo log file group with log sequence number *n*. You can specify any redo log file group that is still online. An error occurs if the log file cannot be found online or the sequence number is not valid. This option can be used to re-archive a log file group.

destination

Specifies the destination device or directory in an operating system. Specification of archive destination devices is installation-specific; see your platform-specific Oracle documentation for examples of specifying archive destinations. On many operating systems, multiple log files can be spooled to the same tape.

If not specified in the command line, the archive destination is derived from the initialization parameter LOG_ARCHIVE_DEST. The command ARCHIVE LOG START *destination* causes the specified device or

directory to become the new default archive destination for all future automatic or manual archives. A destination specified with any other option is a temporary destination that is in effect only for the current (manual) archive. It does not change the default archive destination for subsequent automatic archives. For information about specifying archive destinations, see your platform-specific Oracle documentation.

Usage

You must be connected to an open Oracle database as `SYSOPER`, or `SYSDBA`. For information about connecting to the database, see the `CONNECT` command.

If an online redo log file group fills and none are available for reuse, database operation is suspended. The condition can be resolved by archiving a log file group.

For information about specifying archive destinations, see your platform-specific Oracle documentation.

Note: This command applies only to the current instance. To specify archiving for a different instance or for all instances in a cluster database, use the SQL command `ALTER SYSTEM`. For more information about using SQL commands, see the *Oracle9i SQL Reference*.

Examples

To start up the archiver process and begin automatic archiving, using the archive destination specified in `LOG_ARCHIVE_DEST`, enter



```
ARCHIVE LOG START
```

To stop automatic archiving, enter



```
ARCHIVE LOG STOP
```

To archive the log file group with sequence number 1001 to the destination specified, enter



```
ARCHIVE LOG 1001 '/vobs/oracle/dbs/arch'
```

'arch' specifies the prefix of the filename on the destination device; the remainder of the filename is dependent on the initialization parameter `LOG_ARCHIVE_FORMAT`, which specifies the filename format for archived redo log files.

ATTRIBUTE

Syntax

ATTRIBUTE [*type_name.attribute_name* [*option ...*]]

where *option* represents one of the following clauses:

ALI[AS] *alias*
 CLE[AR]
 FOR[MAT] *format*
 LIKE {*type_name.attribute_name*|*alias*}
 ON|OFF

Specifies display characteristics for a given attribute of an Object Type column, such as format for NUMBER data.

Also lists the current display characteristics for a single attribute or all attributes.

Terms

Enter ATTRIBUTE followed by *type_name.attribute_name* and no other clauses to list the current display characteristics for only the specified attribute. Enter ATTRIBUTE with no clauses to list all current attribute display characteristics.

Refer to the following list for a description of each term or clause:

type_name.attribute_name

Identifies the data item (typically the name of an attribute) within the set of attributes for a given object of Object Type, *type_name*.

If you select objects of the same Object Type, an ATTRIBUTE command for that *type_name.attribute_name* will apply to all such objects you reference in that session.

ALI[AS] *alias*

Assigns a specified alias to a *type_name.attribute_name*, which can be used to refer to the *type_name.attribute_name* in other ATTRIBUTE commands.

CLE[AR]

Resets the display characteristics for the *attribute_name*. The format specification must be a text constant such as A10 or \$9,999—not a variable.

FOR[MAT] *format*

Specifies the display format of the column. The format specification must be a text constant such as A10 or \$9,999—not a variable.

LIKE {*type_name.attribute_name|alias*}

Copies the display characteristics of another attribute. LIKE copies only characteristics not defined by another clause in the current ATTRIBUTE command.

ON|OFF

Controls the status of display characteristics for a column. OFF disables the characteristics for an attribute without affecting the characteristics' definition. ON reinstates the characteristics.

Usage

You can enter any number of ATTRIBUTE commands for one or more attributes. All attribute characteristics set for each attribute remain in effect for the remainder of the session, until you turn the attribute OFF, or until you use the CLEAR COLUMN command. Thus, the ATTRIBUTE commands you enter can control an attribute's display characteristics for multiple SQL SELECT commands.

When you enter multiple ATTRIBUTE commands for the same attribute, SQL*Plus applies their clauses collectively. If several ATTRIBUTE commands apply the same clause to the same attribute, the last one entered will control the output.

Examples

To make the LAST_NAME attribute of the Object Type EMPLOYEE_TYPE 20 characters wide, enter



```
ATTRIBUTE EMPLOYEE_TYPE.LAST_NAME FORMAT A20
```

To format the SALARY attribute of the Object Type EMPLOYEE_TYPE so that it shows millions of dollars, rounds to cents, uses commas to separate thousands, and displays \$0.00 when a value is zero, enter



```
ATTRIBUTE EMPLOYEE_TYPE.SALARY FORMAT $9,999,990.99
```

BREAK

Syntax

BRE[AK] [ON *report_element* [*action* [*action*]]] ...

where:

report_element Requires the following syntax:

{*column*|*expr*|ROW|REPORT}

action Requires the following syntax:

[SKI[P] *n*][SKI[P]] PAGE][NODUP[LICATES]][DUP[LICATES]]

Specifies where and how formatting will change in a report, such as

- suppressing display of duplicate values for a given column
- skipping a line each time a given column value changes
- printing COMPUTED figures each time a given column value changes or at the end of the report (see also the COMPUTE command)

Also lists the current BREAK definition.

Terms

Refer to the following list for a description of each term or clause:

ON *column* [*action* [*action*]]

When you include action(s), specifies action(s) for SQL*Plus to take whenever a break occurs in the specified column (called the *break column*). (*column* cannot have a table or view appended to it. To achieve this, you can alias the column in the SQL statement.) A break is one of three events, a change in the value of a column or expression, the output of a row, or the end of a report

When you omit action(s), BREAK ON *column* suppresses printing of duplicate values in *column* and marks a place in the report where SQL*Plus will perform the computation you specify in a corresponding COMPUTE command.

You can specify `ON column` one or more times. If you specify multiple `ON` clauses, as in

```
BREAK ON DEPARTMENT_ID SKIP PAGE ON JOB_ID -  
SKIP 1 ON SALARY SKIP 1
```

the first `ON` clause represents the *outermost break* (in this case, `ON DEPARTMENT_ID`) and the last `ON` clause represents the *innermost break* (in this case, `ON SALARY`). `SQL*Plus` searches each row of output for the specified break(s), starting with the outermost break and proceeding—in the order you enter the clauses—to the innermost. In the example, `SQL*Plus` searches for a change in the value of `DEPARTMENT_ID`, then `JOB_ID`, then `SALARY`.

Next, `SQL*Plus` executes actions beginning with the action specified for the innermost break and proceeding in reverse order toward the outermost break (in this case, from `SKIP 1` for `ON SALARY` toward `SKIP PAGE` for `ON DEPARTMENT_ID`). `SQL*Plus` executes each action up to and including the action specified for the first occurring break encountered in the initial search.

If, for example, in a given row the value of `JOB_ID` changes—but the values of `DEPARTMENT_ID` and `SALARY` remain the same—`SQL*Plus` skips *two* lines before printing the row (one as a result of `SKIP 1 ON SALARY` and one as a result of `SKIP 1 ON JOB_ID`).

Whenever you use `ON column`, you should also use an `ORDER BY` clause in the `SQL SELECT` command. Typically, the columns used in the `BREAK` command should appear in the same order in the `ORDER BY` clause (although all columns specified in the `ORDER BY` clause need not appear in the `BREAK` command). This prevents breaks from occurring at meaningless points in the report.

If the `BREAK` command specified earlier in this section is used, the following `SELECT` command produces meaningful results:

```
SELECT DEPARTMENT_ID, JOB_ID, SALARY, LAST_NAME  
FROM EMP_DETAILS_VIEW  
WHERE SALARY > 12000  
ORDER BY DEPARTMENT_ID, JOB_ID, SALARY, LAST_NAME;
```

All rows with the same `DEPARTMENT_ID` print together on one page, and within that page all rows with the same `JOB_ID` print in groups. Within each group of jobs, those jobs with the same `SALARY` print in

groups. Breaks in `LAST_NAME` cause no action because `LAST_NAME` does not appear in the `BREAK` command.

`ON expr [action [action]]`

When you include action(s), specifies action(s) for SQL*Plus to take when the value of the expression changes.

When you omit action(s), `BREAK ON expr` suppresses printing of duplicate values of `expr` and marks a place in the report where SQL*Plus will perform the computation you specify in a corresponding `COMPUTE` command.

You can use an expression involving one or more table columns or an alias assigned to a report column in a SQL `SELECT` or SQL*Plus `COLUMN` command. If you use an expression in a `BREAK` command, you must enter `expr` exactly as it appears in the `SELECT` command. If the expression in the `SELECT` command is `a+b`, for example, you cannot use `b+a` or `(a+b)` in a `BREAK` command to refer to the expression in the `SELECT` command.

The information given above for `ON column` also applies to `ON expr`.

`ON ROW [action [action]]`

When you include action(s), specifies action(s) for SQL*Plus to take when a SQL `SELECT` command returns a row. The `ROW` break becomes the innermost break regardless of where you specify it in the `BREAK` command. You should always specify an action when you `BREAK` on a row.

`ON REPORT [action]`

Marks a place in the report where SQL*Plus will perform the computation you specify in a corresponding `COMPUTE` command. Use `BREAK ON REPORT` in conjunction with `COMPUTE` to print grand totals or other “grand” computed values.

The `REPORT` break becomes the outermost break regardless of where you specify it in the `BREAK` command.

Note that SQL*Plus will not skip a page at the end of a report, so you cannot use `BREAK ON REPORT SKIP PAGE`.

Refer to the following list for a description of each action:

SKI[P] *n*

Skips *n* lines before printing the row where the break occurred.

[SKI[P]] PAGE

Skips the number of lines that are defined to be a page before printing the row where the break occurred. The number of lines per page can be set via the PAGESIZE clause of the SET command. Note that PAGESIZE only changes the number of lines that SQL*Plus considers to be a page. Therefore, SKIP PAGE may not always cause a physical page break, unless you have also specified NEWPAGE 0. Note also that if there is a break after the last row of data to be printed in a report, SQL*Plus will not skip the page.

NODUP[LICATES]

Prints blanks rather than the value of a break column when the value is a duplicate of the column's value in the preceding row.

DUP[LICATES]

Prints the value of a break column in every selected row.

Enter BREAK with no clauses to list the current break definition.

Usage

Each new BREAK command you enter replaces the preceding one.

To remove the BREAK command, use CLEAR BREAKS.

Examples

To produce a report that prints duplicate job values, prints the average of SALARY and inserts one blank line when the value of JOB_ID changes, and *additionally* prints the sum of SALARY and inserts another blank line when the value of DEPARTMENT_ID changes, you could enter the following commands. (The example selects departments 50 and 80 and the jobs of clerk and salesman only.)



```
BREAK ON DEPARTMENT_ID SKIP 1 ON JOB_ID SKIP 1 DPLICATES
COMPUTE SUM OF SALARY ON DEPARTMENT_ID
COMPUTE AVG OF SALARY ON JOB_ID
SELECT DEPARTMENT_ID, JOB_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE JOB_ID IN ('SH_CLERK', 'SA_MAN')
```

```
AND DEPARTMENT_ID IN (50, 80)
ORDER BY DEPARTMENT_ID, JOB_ID;
```



DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
50	SH_CLERK	Taylor	3200
	SH_CLERK	Fleaur	3100
	.		
	.		
	.		
	SH_CLERK	Gates	2900

DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
50	SH_CLERK	Perkins	2500
	SH_CLERK	Bell	4000
	.		
	.		
	.		
	SH_CLERK	Grant	2600
	*****		-----
	avg		3215

DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
*****			-----
	sum		64300
80	SA_MAN	Russell	14000
	SA_MAN	Partners	13500
	SA_MAN	Errazuriz	12000
	SA_MAN	Cambrault	11000
	SA_MAN	Zlotkey	10500
	*****		-----
	avg		12200

DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
*****			-----
	sum		61000

25 rows selected.

BTITLE

Syntax

```
BTI[TLE] [printspec [text|variable] ...] [ON|OFF]
```

Places and formats a specified title at the bottom of each report page, or lists the current BTITLE definition.

Terms

Refer to the TTITLE command for additional information on terms and clauses in the BTITLE command syntax.

Enter BTITLE with no clauses to list the current BTITLE definition.

Usage

If you do not enter a *printspec* clause before the first occurrence of *text*, BTITLE left justifies the text. SQL*Plus interprets BTITLE in the new form if a valid *printspec* clause (LEFT, SKIP, COL, and so on) immediately follows the command name.

Examples

To set a bottom title with CORPORATE PLANNING DEPARTMENT on the left and a date on the right, enter



```
BTITLE LEFT 'CORPORATE PLANNING DEPARTMENT' -  
RIGHT '1 JAN 2001'
```

To set a bottom title with CONFIDENTIAL in column 50, followed by six spaces and a date, enter



```
BTITLE COL 50 'CONFIDENTIAL' TAB 6 '1 JAN 2001'
```

CLEAR

Syntax

CL[EAR] *option* ...

where *option* represents one of the following clauses:

BRE[AKS]
BUFF[ER]
COL[UMNS]
COMP[UTES]
SQL
TIMI[NG]

Resets or erases the current value or setting for the specified option.

Terms

Refer to the following list for a description of each term or clause:

BRE[AKS]

Removes the break definition set by the BREAK command.

BUFF[ER]

Clears text from the buffer. CLEAR BUFFER has the same effect as CLEAR SQL, unless you are using multiple buffers.

COL[UMNS]

Resets column display attributes set by the COLUMN command to default settings for all columns. To reset display attributes for a single column, use the CLEAR clause of the COLUMN command. CLEAR COLUMNS also clears the ATTRIBUTEs for that column.

COMP[UTES]

Removes all COMPUTE definitions set by the COMPUTE command.

SQL

Clears the text from SQL buffer. CLEAR SQL has the same effect as CLEAR BUFFER, unless you are using multiple buffers.

TIMI[NG]

Deletes all timers created by the TIMING command.

Examples



To clear breaks, enter

```
CLEAR BREAKS
```



To clear column definitions, enter

```
CLEAR COLUMNS
```

COLUMN

Syntax

COL[UMN] [{*column*|*expr*} [*option* ...]]

where *option* represents one of the following clauses:

ALI[AS] *alias*
 CLE[AR]
 ENTMAP {ON|OFF}
 FOLD_A[FTER]
 FOLD_B[EFORE]
 FOR[MAT] *format*
 HEA[DING] *text*
 JUS[TIFY] {L[EFT]|C[ENTER]|C[ENTRE]|R[IGHT]}
 LIKE {*expr*|*alias*}
 NEWL[INE]
 NEW_V[ALUE] *variable*
 NOPRI[NT]|PRI[NT]
 NUL[L] *text*
 OLD_V[ALUE] *variable*
 ON|OFF
 WRA[PPED]|WOR[D_WRAPPED]|TRU[NCATED]

Specifies display attributes for a given column, such as

- text for the column heading
- alignment of the column heading
- format for NUMBER data
- wrapping of column data

Also lists the current display attributes for a single column or all columns.

Terms

Enter COLUMN followed by *column* or *expr* and no other clauses to list the current display attributes for only the specified column or expression. Enter COLUMN with no clauses to list all current column display attributes.

Refer to the following list for a description of each term or clause:

{column|expr}

Identifies the data item (typically, the name of a column) in a SQL SELECT command to which the column command refers. If you use an expression in a COLUMN command, you must enter *expr* exactly as it appears in the SELECT command. If the expression in the SELECT command is *a+b*, for example, you cannot use *b+a* or *(a+b)* in a COLUMN command to refer to the expression in the SELECT command.

If you select columns with the same name from different tables, a COLUMN command for that column name will apply to both columns. That is, a COLUMN command for the column `LAST_NAME` applies to all columns named `LAST_NAME` that you reference in this session. COLUMN ignores table name prefixes in SELECT commands. Also, spaces are ignored unless the name is placed in double quotes.

To format the columns differently, assign a unique alias to each column within the SELECT command itself (do not use the ALIAS clause of the COLUMN command) and enter a COLUMN command for each column's alias.

ALI[AS] *alias*

Assigns a specified *alias* to a column, which can be used to refer to the column in BREAK, COMPUTE, and other COLUMN commands.

CLE[AR]

Resets the display attributes for the column to default values.

To reset the attributes for all columns, use the CLEAR COLUMNS command. CLEAR COLUMNS also clears the ATTRIBUTES for that column.

ENTMAP {ON|OFF}

Allows entity mapping to be turned on or off for selected columns in HTML output. This feature allows you to include, for example, HTML hyperlinks in a column of data, while still mapping entities in other columns of the same report. By turning entity mapping off for a column containing HTML hyperlinks, the HTML anchor tag delimiters, `<`, `>`, `"` and `&`, are correctly interpreted in the report. Otherwise they would be replaced with their respective entities, `<`, `>`, `"` and `&`, preventing web browsers from correctly interpreting the HTML.

Entities in the column heading and any COMPUTE labels or output appearing in the column are mapped or not mapped according to the value of ENTMAP for the column.

The default setting for COLUMN ENTMAP is the current setting of the MARKUP HTML ENTMAP option. For more information about the MARKUP HTML ENTMAP option, see MARKUP Options in Chapter 7 and SET later this Chapter.

FOLD_A[FTER]

Inserts a carriage return after the column heading and after each row in the column. SQL*Plus does not insert an extra carriage return after the last column in the SELECT list.

FOLD_B[EFORE]

Inserts a carriage return before the column heading and before each row of the column. SQL*Plus does not insert an extra carriage return before the first column in the SELECT list.

FOR[MAT] *format*

Specifies the display format of the column. The format specification must be a text constant such as A10 or \$9,999—not a variable.

Character Columns The default width of CHAR, NCHAR, VARCHAR2 (VARCHAR) and NVARCHAR2 (NCHAR VARYING) columns is the width of the column in the database. SQL*Plus formats these datatypes left-justified. If a value does not fit within the column width, SQL*Plus wraps or truncates the character string depending on the setting of SET WRAP.

A LONG, CLOB or NCLOB column's width defaults to the value of SET LONGCHUNKSIZE or SET LONG, whichever one is smaller.

To change the width of a datatype to *n*, use FORMAT An. (A stands for alphanumeric.) If you specify a width shorter than the column heading, SQL*Plus truncates the heading. If you specify a width for a LONG, CLOB, or NCLOB column, SQL*Plus uses the LONGCHUNKSIZE or the specified width, whichever is smaller, as the column width.

DATE Columns The default width and format of unformatted DATE columns in SQL*Plus is derived from the NLS parameters in effect. Otherwise, the default width is A9. In Oracle9i, the NLS parameters may be set in your database parameter file or may be environment variables or an equivalent platform-specific mechanism. They may also be specified

for each session with the ALTER SESSION command. (See the documentation for Oracle9i for a complete description of the NLS parameters).

You can change the format of any DATE column using the SQL function TO_CHAR in your SQL SELECT statement. You may also wish to use an explicit COLUMN FORMAT command to adjust the column width.

When you use SQL functions like TO_CHAR, Oracle automatically allows for a very wide column.

To change the width of a DATE column to *n*, use the COLUMN command with FORMAT An. If you specify a width shorter than the column heading, the heading is truncated.

NUMBER Columns To change a NUMBER column's width, use FORMAT followed by an element as specified in Table 5-1.

Table 5-1 Number Formats

Element	Examples	Description
9	9999	Number of "9"s specifies number of significant digits returned. Blanks are displayed for leading zeroes. A zero (0) is displayed for a value of zero.
0	0999 9990	Displays a leading zero or a value of zero in this position as 0.
\$	\$9999	Prefixes value with dollar sign.
B	B9999	Displays a zero value as blank, regardless of "0"s in the format model.
MI	9999MI	Displays "-" after a negative value. For a positive value, a trailing space is displayed.
S	S9999	Returns "+" for positive values and "-" for negative values in this position.
PR	9999PR	Displays a negative value in <angle brackets>. For a positive value, a leading and trailing space is displayed.
D	99D99	Displays the decimal character in this position, separating the integral and fractional parts of a number.
G	9G999	Displays the group separator in this position.

Table 5–1 Number Formats

Element	Examples	Description
C	C999	Displays the ISO currency symbol in this position.
L	L999	Displays the local currency symbol in this position.
, (comma)	9,999	Displays a comma in this position.
. (period)	99.99	Displays a period (decimal point) in this position, separating the integral and fractional parts of a number.
V	999V99	Multiplies value by 10^n , where n is number of “9”s after “V”.
EEEE	9.999EEEE	Displays value in scientific notation (format must contain exactly four "E"s).
RN or rn	RN	Displays upper- or lowercase Roman numerals. Value can be an integer between 1 and 3999.
DATE	DATE	Displays value as a date in MM/DD/YY format; used to format NUMBER columns that represent Julian dates.

The MI and PR format elements can only appear in the last position of a number format model. The S format element can only appear in the first or last position.

If a number format model does not contain the MI, S or PR format elements, negative return values automatically contain a leading negative sign and positive values automatically contain a leading space.

A number format model can contain only a single decimal character (D) or period (.), but it can contain multiple group separators (G) or commas (,). A group separator or comma cannot appear to the right of a decimal character or period in a number format model.

SQL*Plus formats NUMBER data right-justified. A NUMBER column's width equals the width of the heading or the width of the FORMAT plus one space for the sign, whichever is greater. If you do not explicitly use FORMAT, then the column's width will always be at least the value of SET NUMWIDTH.

SQL*Plus may round your NUMBER data to fit your format or field width.

If a value cannot fit within the column width, SQL*Plus indicates overflow by displaying a pound sign (#) in place of each digit the width allows.

If a positive value is extremely large and a numeric overflow occurs when rounding a number, then the infinity sign (~) replaces the value. Likewise, if a negative value is extremely small and a numeric overflow occurs when rounding a number, then the negative infinity sign replaces the value (-~).

HEA[DING] *text*

Defines a column heading. If you do not use a HEADING clause, the column's heading defaults to *column* or *expr*. If *text* contains blanks or punctuation characters, you must enclose it with single or double quotes. Each occurrence of the HEADSEP character (by default, "|") begins a new line.

For example,

```
COLUMN LAST_NAME HEADING 'Employee |Name'
```

would produce a two-line column heading. See the HEADSEP variable of the SET command in this chapter for information on changing the HEADSEP character.

JUS[TIFY] {L[EFT]|C[ENTER]|C[ENTRE]|R[IGHT]}

Aligns the heading. If you do not use a JUSTIFY clause, headings for NUMBER columns default to RIGHT and headings for other column types default to LEFT.

LIKE {*expr*|*alias*}

Copies the display attributes of another column or expression (whose attributes you have already defined with another COLUMN command). LIKE copies only attributes not defined by another clause in the current COLUMN command.

NEWL[INE]

Starts a new line before displaying the column's value. NEWLINE has the same effect as FOLD_BEFORE.

NEW_V[ALUE] *variable*

Specifies a variable to hold a column value. You can reference the variable in TTITLE commands. Use NEW_VALUE to display column values or the date in the top title. You must include the column in a

BREAK command with the SKIP PAGE action. The variable name cannot contain a pound sign (#).

NEW_VALUE is useful for master/detail reports in which there is a new master record for each page. For master/detail reporting, you must also include the column in the ORDER BY clause. See the example at the end of this command description.

For information on displaying a column value in the bottom title, see COLUMN OLD_VALUE. For more information on referencing variables in titles, see the TTITLE command later in this chapter. For information on formatting and valid format models, see the FORMAT command.

NOPRI[NT]|PRI[NT]

Controls the printing of the column (the column heading and all the selected values). NOPRINT turns off the screen output and printing of the column. PRINT turns the printing of the column on.

NUL[L] *text*

Controls the text SQL*Plus displays for null values in the given column. The default is a white space. SET NULL controls the text displayed for all null values for all columns, unless overridden for a specific column by the NULL clause of the COLUMN command. When a NULL value is SELECTed, a variable's type will always become CHAR so the SET NULL text can be stored in it.

OLD_V[ALUE] *variable*

Specifies a variable to hold a column value. You can reference the variable in BTITLE commands. Use OLD_VALUE to display column values in the bottom title. You must include the column in a BREAK command with the SKIP PAGE action.

OLD_VALUE is useful for master/detail reports in which there is a new master record for each page. For master/detail reporting, you must also include the column in the ORDER BY clause.

For information on displaying a column value in the top title, see COLUMN NEW_VALUE. For more information on referencing variables in titles, see the TTITLE command later in this chapter.

ON|OFF

Controls the status of display attributes for a column. OFF disables the attributes for a column without affecting the attributes' definition. ON reinstates the attributes.

WRA[PPED]|WOR[D_WRAPPED]|TRU[NCATED]

Specifies how SQL*Plus will treat a datatype or DATE string that is too wide for a column. WRAPPED wraps the string within the column bounds, beginning new lines when required. When WORD_WRAP is enabled, SQL*Plus left justifies each new line, skipping all leading whitespace (for example, returns, newline characters, tabs and spaces), including embedded newline characters. Embedded whitespace not on a line boundary is not skipped. TRUNCATED truncates the string at the end of the first line of display.

Usage

You can enter any number of COLUMN commands for one or more columns. All column attributes set for each column remain in effect for the remainder of the session, until you turn the column OFF, or until you use the CLEAR COLUMN command. Thus, the COLUMN commands you enter can control a column's display attributes for multiple SQL SELECT commands.

When you enter multiple COLUMN commands for the same column, SQL*Plus applies their clauses collectively. If several COLUMN commands apply the same clause to the same column, the last one entered will control the output.

Examples

To make the LAST_NAME column 20 characters wide and display EMPLOYEE NAME on two lines as the column heading, enter



```
COLUMN LAST_NAME FORMAT A20 HEADING 'EMPLOYEE|NAME'
```

To format the SALARY column so that it shows millions of dollars, rounds to cents, uses commas to separate thousands, and displays \$0.00 when a value is zero, enter



```
COLUMN SALARY FORMAT $9,999,990.99
```

To assign the alias NET to a column containing a long expression, to display the result in a dollar format, and to display <NULL> for null values, you might enter



```
COLUMN SALARY+COMMISSION_PCT+BONUS-EXPENSES-INS-TAX ALIAS NET  
COLUMN NET FORMAT $9,999,999.99 NULL '<NULL>'
```

Note that the example divides this column specification into two commands. The first defines the alias NET, and the second uses NET to define the format.

Also note that in the first command you must enter the expression exactly as you enter it in the SELECT command. Otherwise, SQL*Plus cannot match the COLUMN command to the appropriate column.

To wrap long values in a column named REMARKS, you can enter



```
COLUMN REMARKS FORMAT A20 WRAP
```



```
CUSTOMER  DATE          QUANTITY REMARKS
-----
123       25-AUG-2001      144 This order must be s
             shipped by air freight
             t to ORD
```

If you replace WRAP with WORD_WRAP, REMARKS looks like this:

```
CUSTOMER  DATE          QUANTITY REMARKS
-----
123       25-AUG-2001      144 This order must be
             shipped by air freight
             to ORD
```

If you specify TRUNCATE, REMARKS looks like this:

```
CUSTOMER  DATE          QUANTITY REMARKS
-----
123       25-AUG-2001      144 This order must be s
```

In order to print the current date and the name of each job in the top title, enter the following. Use the EMPLOYEES table of the HR schema in this case instead of EMP_DETAILS_VIEW as you have used up to now. (For details on creating a date variable, see "Displaying the Current Date in Titles" under "Defining Page and Report Titles and Dimensions" in Chapter 4.)



```
COLUMN JOB_ID NOPRINT NEW_VALUE JOBVAR
COLUMN TODAY  NOPRINT NEW_VALUE DATEVAR
BREAK ON JOB_ID SKIP PAGE ON TODAY
TTITLE CENTER 'Job Report' RIGHT DATEVAR SKIP 2 -
LEFT 'Job:      ' JOBVAR SKIP 2
SELECT TO_CHAR(SYSDATE, 'MM/DD/YYYY') TODAY,
LAST_NAME, JOB_ID, MANAGER_ID, HIRE_DATE, SALARY, DEPARTMENT_ID
FROM EMPLOYEES WHERE JOB_ID IN ('MK_MAN', 'SA_MAN')
ORDER BY JOB_ID, LAST_NAME;
```

Your two page report would look similar to the following report, with “Job Report” centered within your current linesize:



```

                                Job Report
04/19/01

Job:      MK_MAN

LAST
NAME                MANAGER_ID HIRE_DATE          SALARY DEPARTMENT_ID
-----
Hartstein            100 17-FEB-96      $13,000.00          20
                                -----
                                $13,000.00
    
```

```

                                Job Report                                04/19/01

Job:      SA_MAN

LAST
NAME                MANAGER_ID HIRE_DATE          SALARY DEPARTMENT_ID
-----
Errazuriz            100 10-MAR-97      $12,000.00          80
Zlotkey               100 29-JAN-00      $10,500.00          80
Cambrault             100 15-OCT-99      $11,000.00          80
Russell               100 01-OCT-96      $14,000.00          80
Partners              100 05-JAN-97      $13,500.00          80
                                -----
    
```

```

                                Job Report                                04/19/01

Job:      SA_MAN

LAST
NAME                MANAGER_ID HIRE_DATE          SALARY DEPARTMENT_ID
-----
                                -----
                                $12,200.00
    
```

6 rows selected.

To change the default format of DATE columns to 'YYYY-MM-DD', you can enter



```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD';
```



Session altered.

To display the change, enter a SELECT statement, such as:



```
SELECT HIRE_DATE
FROM EMPLOYEES
WHERE EMPLOYEE_ID = 206;
```



Job Report

04/19/01

```
Job:      SA_MAN
```

```
HIRE_DATE
-----
1994-06-07
```

See the *Oracle9i SQL Reference* for information on the ALTER SESSION command.

Note that in a SELECT statement, some SQL calculations or functions, such as TO_CHAR, may cause a column to be very wide. In such cases, use the FORMAT option to alter the column width.

COMPUTE

Syntax

```
COMP[UTE] [function [LAB[EL] text] ...
  OF {expr|column|alias} ...
  ON {expr|column|alias|REPORT|ROW} ...]
```

Calculates and prints summary lines, using various standard computations, on subsets of selected rows. It also lists all COMPUTE definitions.

Terms

Refer to the following list for a description of each term or clause:

function ...

Represents one of the functions listed in Table 5–2. If you specify more than one function, use spaces to separate the functions.

COMPUTE command functions are always executed in the sequence AVG, COUNT, MINIMUM, MAXIMUM, NUMBER, SUM, STD, VARIANCE, regardless of their order in the COMPUTE command.

Table 5–2 COMPUTE Functions

Function	Computes	Applies to Datatypes
AVG	Average of non-null values	NUMBER
COU[NT]	Count of non-null values	all types
MIN[IMUM]	Minimum value	NUMBER, CHAR, NCHAR, VARCHAR2 (VARCHAR), NVARCHAR2 (NCHAR VARYING)
MAX[IMUM]	Maximum value	NUMBER, CHAR, NCHAR, VARCHAR2 (VARCHAR), NVARCHAR2 (NCHAR VARYING)
NUM[BER]	Count of rows	all types
SUM	Sum of non-null values	NUMBER

Table 5–2 COMPUTE Functions

Function	Computes	Applies to Datatypes
STD	Standard deviation of non-null values	NUMBER
VAR[IANCE]	Variance of non-null values	NUMBER

LAB[EL] *text*

Defines the label to be printed for the computed value. If no LABEL clause is used, *text* defaults to the unabbreviated function keyword. You must place single quotes around *text* containing spaces or punctuation. The label prints left justified and truncates to the column width or line-size, whichever is smaller. The maximum label length is 500 characters.

The label for the computed value appears in the break column specified. To suppress the label, use the NOPRINT option of the COLUMN command on the break column.

If you repeat a function in a COMPUTE command, SQL*Plus issues a warning and uses the first occurrence of the function.

With ON REPORT and ON ROW computations, the label appears in the first column listed in the SELECT statement. The label can be suppressed by using a NOPRINT column first in the SELECT statement. When you compute a function of the first column in the SELECT statement ON REPORT or ON ROW, then the computed value appears in the first column and the label is not displayed. To see the label, select a dummy column first in the SELECT list.

OF {*expr*|*column*|*alias*} ...

In the OF clause, you can refer to an expression or function reference in the SELECT statement by placing the expression or function reference in double quotes. Column names and aliases do not need quotes.

ON {*expr*|*column*|*alias*|REPORT|ROW} ...

Specifies the event SQL*Plus will use as a break. (*column* cannot have a table or view appended to it. To achieve this, you can alias the column in the SQL statement.) COMPUTE prints the computed value and restarts the computation when the event occurs (that is, when the value of the expression changes, a new ROW is fetched, or the end of the report is reached).

If multiple COMPUTE commands reference the same column in the ON clause, only the last COMPUTE command applies.

To reference a SQL SELECT expression or function reference in an ON clause, place the expression or function reference in quotes. Column names and aliases do not need quotes.

Enter COMPUTE without clauses to list all COMPUTE definitions.

Usage

In order for the computations to occur, the following conditions must all be true:

- One or more of the expressions, columns, or column aliases you reference in the OF clause must also be in the SELECT command.
- The expression, column, or column alias you reference in the ON clause must occur in the SELECT command and in the most recent BREAK command.
- If you reference either ROW or REPORT in the ON clause, also reference ROW or REPORT in the most recent BREAK command.

To remove all COMPUTE definitions, use the CLEAR COMPUTES command.

Examples

To subtotal the salary for the “account manager” and “salesman” job classifications with a compute label of “TOTAL”, enter



```
BREAK ON JOB_ID SKIP 1;
COMPUTE SUM LABEL 'TOTAL' OF SALARY ON JOB_ID;
SELECT JOB_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE JOB_ID IN ('AC_MGR', 'SA_MAN')
ORDER BY JOB_ID, SALARY;
```



JOB_ID	LAST_NAME	SALARY
AC_MGR	Higgins	12000
*****		-----
TOTAL		12000
SA_MAN	Zlotkey	10500
	Cambrault	11000
	Errazuriz	12000
	Partners	13500
	Russell	14000


```
*****
TOTAL                                -----
                                      61000
```

6 rows selected.

To calculate the total of salaries greater than 12,000 on a report, enter



```
COMPUTE SUM OF SALARY ON REPORT
BREAK ON REPORT
COLUMN DUMMY HEADING ''
SELECT ' ' DUMMY, SALARY, EMPLOYEE_ID
FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000
ORDER BY SALARY;
```



```

          SALARY EMPLOYEE_ID
-----
          13000         201
          13500         146
          14000         145
          17000         101
          17000         102
          24000         100
-----
sum          98500
```

6 rows selected.

To calculate the average and maximum salary for the executive and accounting departments, enter



```
BREAK ON DEPARTMENT_NAME SKIP 1
COMPUTE AVG LABEL 'Dept Average' -
          MAX LABEL 'Dept Maximum' -
          OF SALARY ON DEPARTMENT_NAME
SELECT DEPARTMENT_NAME, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE DEPARTMENT_NAME IN ('Executive', 'Accounting')
ORDER BY DEPARTMENT_NAME;
```



```
DEPARTMENT_NAME          LAST_NAME          SALARY
-----
Accounting                Higgins            12000
                          Gietz              8300
*****
```

COMPUTE

Dept Average		10150
Dept Maximum		12000
Executive	King	24000
	Kochhar	17000
	De Haan	17000

Dept Average		19333.3333
Dept Maximum		24000

To sum salaries for departments <= 20 without printing the compute label, enter



```
COLUMN DUMMY NOPRINT
COMPUTE SUM OF SALARY ON DUMMY
BREAK ON DUMMY SKIP 1
SELECT DEPARTMENT_ID DUMMY, DEPARTMENT_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE DEPARTMENT_ID <= 20
ORDER BY DEPARTMENT_ID;
```



DEPARTMENT_ID	LAST_NAME	SALARY
10	Whalen	4400

		4400
20	Hartstein	13000
20	Fay	6000

		19000

To total the salary at the end of the report without printing the compute label, enter



```
COLUMN DUMMY NOPRINT
COMPUTE SUM OF SALARY ON DUMMY
BREAK ON DUMMY
SELECT NULL DUMMY, DEPARTMENT_ID, LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE DEPARTMENT_ID <= 30
ORDER BY DEPARTMENT_ID;
```



DEPARTMENT_ID	LAST_NAME	SALARY
10	Whalen	4400
20	Hartstein	13000
20	Fay	6000

30	Raphaely	11000
30	Khoo	3100
30	Baida	2900
30	Tobias	2800
30	Himuro	2600
30	Colmenares	2500

		48300

9 rows selected.

CONNECT

Syntax

CONN[ECT] { *logon* / } [AS {SYSOPER|SYSDBA}]

where *logon* requires the following syntax:

username/*password*[@*connect_identifier*]

Connects a given username to Oracle.

Terms

Refer to the following list for a description of each term or clause:

username/*password*

Represent the username and password with which you wish to connect to Oracle. If you enter a slash (/) or simply enter Return to the prompt for *username*, SQL*Plus logs you in using a default logon (see “/(slash)” below).

connect_identifier

Consists of an Oracle Net connect identifier. The exact syntax depends upon the Oracle Net communications protocol your Oracle installation uses. For more information, refer to the Oracle Net manual appropriate for your protocol or contact your DBA. SQL*Plus does not prompt for a service name, but uses your default database if you do not include a specification.

/(slash)

Represents a default logon using operating system authentication. You cannot enter a *connect_identifier* if you use a default logon. In a default logon, SQL*Plus typically attempts to log you in using the username OPS\$name, where *name* is your operating system username. See the *Oracle9i Administrator's Guide* for information about operating system authentication.

AS {SYSOPER|SYSDBA}

The AS clause allows privileged connections by users who have been granted SYSOPER or SYSDBA system privileges. You can use either of these privileged connections with the default logon, /. For information about system privileges, see the *Oracle9i Administrator's Guide*.

Usage

CONNECT commits the current transaction to the database, disconnects the current username from Oracle, and reconnects with the specified username.

If you log on or connect as a user whose account has expired, SQL*Plus prompts you to change your password before you can connect.

If an account is locked, a message is displayed and connection into that account (as that user) is not permitted until the account is unlocked by your DBA.

For more information about user account management, refer to the CREATE and ALTER USER commands, and the CREATE PROFILE command in the *Oracle9i SQL Reference*.

Examples

To connect across Oracle Net with username HR and password HR to the database known by the Oracle Net alias as FLEETDB, enter



```
CONNECT HR/HR@FLEETDB
```

To connect as user HR, letting SQL*Plus prompt you for the password, enter



```
CONNECT HR
```

For more information about setting up your password file, refer to the *Oracle9i Administrator's Guide*.

To use a password file to connect to an instance on the current node as a privileged user named HR with the password HR, enter



```
CONNECT HR/HR AS SYSDBA
```

To connect to an instance on the current node as a privileged default user, enter



```
CONNECT / AS SYSDBA
```

In the last two examples, your default schema becomes SYS.

COPY

Syntax

```
COPY {FROM database | TO database | FROM database TO database}  
{APPEND|CREATE|INSERT|REPLACE} destination_table [(column, column, column, ...)]  
USING query
```

where *database* has the following syntax:

```
username[/password]@connect_identifier
```

Copies data from a query to a table in a local or remote database. COPY supports the following datatypes:

```
CHAR  
DATE  
LONG  
NUMBER  
VARCHAR2
```

The COPY command is not being enhanced to handle datatypes or features introduced with, or after Oracle8. The COPY command is likely to be made obsolete in a future release.

Terms

Refer to the following list for a description of each term or clause:

FROM database

Specifies the database that contains the data to be copied. If you omit the FROM clause, the source defaults to the database to which SQL*Plus is connected (that is, the database that other commands address). You must include a FROM clause to specify a source database other than the default.

TO database

Specifies the database containing the destination table. If you omit the TO clause, the destination defaults to the database to which SQL*Plus is connected (that is, the database that other commands address). You must include a TO clause to specify a destination database other than the default.

database

Specifies *username[/password] @connect_identifier* of the Oracle source or destination database you wish to COPY FROM or COPY TO. If you do not specify *password* in either the FROM clause or the TO clause, SQL*Plus will prompt you for it. SQL*Plus suppresses the display of your password response.

You must include the *connect_identifier* clause which consists of an Oracle Net connection string, to specify the source or destination database. The exact syntax depends upon the Oracle Net communications protocol your Oracle installation uses. For more information, refer to the Oracle Net manual appropriate for your protocol or contact your DBA.

APPEND

Inserts the rows from *query* into *destination_table* if the table exists. If *destination_table* does not exist, COPY creates it.

CREATE

Inserts the rows from *query* into *destination_table* after first creating the table. If *destination_table* already exists, COPY returns an error.

INSERT

Inserts the rows from *query* into *destination_table*. If *destination_table* does not exist, COPY returns an error. When using INSERT, the USING *query* must select one column for each column in the *destination_table*.

REPLACE

Replaces *destination_table* and its contents with the rows from *query*. If *destination_table* does not exist, COPY creates it. Otherwise, COPY drops the existing table and replaces it with a table containing the copied data.

destination_table

Represents the table you wish to create or to which you wish to add data.

(column, column, column, ...)

Specifies the names of the columns in *destination_table*. You must enclose a name in double quotes if it contains lowercase letters or blanks.

If you specify columns, the number of columns must equal the number of columns selected by the query. If you do not specify any columns, the copied columns will have the same names in the destination table as they had in the source if COPY creates *destination_table*.

USING *query*

Specifies a SQL query (SELECT command) determining which rows and columns COPY copies.

Usage

To enable the copying of data between Oracle and non-Oracle databases, NUMBER columns are changed to DECIMAL columns in the destination table. Hence, if you are copying between Oracle databases, a NUMBER column with no precision will be changed to a DECIMAL(38) column. When copying between Oracle databases, you should use SQL commands (CREATE TABLE AS and INSERT) or you should ensure that your columns have a precision specified.

The SQL*Plus SET variable LONG limits the length of LONG columns that you copy. If any LONG columns contain data longer than the value of LONG, COPY truncates the data.

SQL*Plus performs a commit at the end of each successful COPY. If you set the SQL*Plus SET variable COPYCOMMIT to a positive value *n*, SQL*Plus performs a commit after copying every *n* batches of records. The SQL*Plus SET variable ARRAYSIZE determines the size of a batch.

Some operating environments require that service names be placed in double quotes.

Examples

The following command copies the entire EMPLOYEES table to a table named WESTEMPLOYEES. Note that the tables are located in two different databases. If WESTEMPLOYEES already exists, SQL*Plus replaces the table and its contents. The columns in WESTEMPLOYEES have the same names as the columns in the source table, EMPLOYEES.



```
COPY FROM HR/HR@HQ TO JOHN/CHROME@WEST -  
REPLACE WESTEMPLOYEES -  
USING SELECT * FROM EMPLOYEES
```

The following command copies selected records from EMPLOYEES to the database to which SQL*Plus is connected. SQL*Plus creates SALESMEN through the copy.

SQL*Plus copies only the columns EMPLOYEE_ID and LAST_NAME, and at the destination names them EMPLOYEE_ID and SA_MAN.



```
COPY FROM HR/HR@ORACLE01 -  
CREATE SALESMEN (EMPLOYEE_ID, SA_MAN) -  
USING SELECT EMPLOYEE_ID, LAST_NAME FROM EMPLOYEES -  
WHERE JOB_ID='SA_MAN' ;
```

DEFINE

Syntax

DEF[INE] [*variable*][*variable = text*]

Specifies a user variable and assigns it a CHAR value, or lists the value and variable type of a single variable or all variables.

Terms

Refer to the following list for a description of each term or clause:

variable

Represents the user variable whose value you wish to assign or list.

text

Represents the CHAR value you wish to assign to *variable*. Enclose *text* in single quotes if it contains punctuation or blanks.

variable = text

Defines (names) a user variable and assigns it a CHAR value.

Enter DEFINE followed by *variable* to list the value and type of *variable*. Enter DEFINE with no clauses to list the values and types of all user variables.

Usage

Defined variables retain their values until one of the following events occurs:

- you enter a new DEFINE command referencing the variable
- you enter an UNDEFINE command referencing the variable
- you enter an ACCEPT command referencing the variable
- you reference the variable in the NEW_VALUE or OLD_VALUE clause of the COLUMN command and reference the column in a subsequent SQL SELECT command
- you EXIT SQL*Plus

Whenever you run a stored query or command file, SQL*Plus substitutes the value of *variable* for each substitution variable referencing *variable* (in the form *&variable* or *&&variable*). SQL*Plus will not prompt you for the value of *variable* in this session until you UNDEFINE *variable*.

Note that you can use DEFINE to define the variable, `_EDITOR`, which establishes the host system editor invoked by the SQL*Plus EDIT command.

If you continue the value of a defined variable on multiple lines (using the SQL*Plus command continuation character), SQL*Plus replaces each continuation character and carriage return you enter with a space in the resulting variable. For example, SQL*Plus interprets

```
DEFINE TEXT = 'ONE-
TWO-
THREE'
```

as

```
DEFINE TEXT = 'ONE TWO THREE'
```

You should avoid defining variables with names that may be identical to values that you will pass to them, as unexpected results can occur. If a value supplied for a defined variable matches a variable name, then the contents of the matching variable are used instead of the supplied value.

Some variables are predefined when SQL*Plus starts. Enter DEFINE to see their definitions.

Examples



To assign the value `MANAGER` to the variable `POS`, type:

```
DEFINE POS = MANAGER
```

If you execute a command that contains a reference to `&POS`, SQL*Plus will substitute the value `MANAGER` for `&POS` and will not prompt you for a `POS` value.



To assign the `CHAR` value `20` to the variable `DEPARTMENT_ID`, type:

```
DEFINE DEPARTMENT_ID = 20
```

Even though you enter the number `20`, SQL*Plus assigns a `CHAR` value to `DEPARTMENT_ID` consisting of two characters, `2` and `0`.

To list the definition of `DEPARTMENT_ID`, enter



```
DEFINE DEPARTMENT_ID
DEFINE DEPARTMENT_ID = "20" (CHAR)
```

This result shows that the value of `DEPARTMENT_ID` is `20`.

DESCRIBE

Syntax

DESC[RIBE] {[*schema*.]*object*[@*connect_identifier*]}

Lists the column definitions for the specified table, view, or synonym or the specifications for the specified function or procedure.

Terms

Refer to the following list for a description of each term or clause:

schema

Represents the schema where the *object* resides. If you omit *schema*, SQL*Plus assumes you own *object*.

object

Represents the table, view, type, procedure, function, package or synonym you wish to describe.

@connect_identifier

Consists of the database link name corresponding to the database where *object* exists. For more information on which privileges allow access to another table in a different schema, refer to the *Oracle9i SQL Reference*.

Usage

The description for tables, views, types and synonyms contains the following information:

- each column's name
- whether or not null values are allowed (NULL or NOT NULL) for each column
- datatype of columns, for example, NUMBER, CHAR, VARCHAR2 (VARCHAR), LONG, DATE, RAW, LONGRAW, or ROWID
- precision of columns (and scale, if any, for a numeric column)

When you do a DESCRIBE, VARCHAR columns are returned with a type of VARCHAR2.

The DESCRIBE command allows you to describe objects recursively to the depth level set in the SET DESCRIBE command. You can also display the line number and

indentation of the attribute or column name when an object contains multiple object types. For more information, see the SET command later in this chapter.

To control the width of the data displayed, use the SET LINESIZE command. For more information, see the SET command later in this chapter.

The description for functions and procedures contains the following information:

- the type of PL/SQL object (function or procedure)
- the name of the function or procedure
- the type of value returned (for functions)
- the argument names, types, whether input or output, and default values, if any

Examples

To describe the view EMP_DETAILS_VIEW, enter



```
DESCRIBE EMP_DETAILS_VIEW
```



Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
JOB_ID	NOT NULL	VARCHAR2(10)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)
LOCATION_ID		NUMBER(4)
COUNTRY_ID		CHAR(2)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
JOB_TITLE	NOT NULL	VARCHAR2(35)
CITY	NOT NULL	VARCHAR2(30)
STATE_PROVINCE		VARCHAR2(25)
COUNTRY_NAME		VARCHAR2(40)
REGION_NAME		VARCHAR2(25)

To describe a procedure called CUSTOMER_LOOKUP, enter



```
DESCRIBE customer_lookup
```



```
PROCEDURE customer_lookup
Argument Name      Type      In/Out  Default?
```

```

-----
CUST_ID          NUMBER    IN
CUST_NAME        VARCHAR2  OUT

```

To create and describe the package **APACK** that contains the procedures **aproc** and **bproc**, enter



```

CREATE PACKAGE apack AS
PROCEDURE aproc(P1 CHAR, P2 NUMBER);
PROCEDURE bproc(P1 CHAR, P2 NUMBER);
END apack;
/

```



Package created.



DESCRIBE apack



```

PROCEDURE APROC
Argument Name          Type          In/Out Default?
-----
P1                     CHAR          IN
P2                     NUMBER        IN
PROCEDURE BPROC
Argument Name          Type          In/Out Default?
-----
P1                     CHAR          IN
P2                     NUMBER        IN

```

To create and describe the object type **ADDRESS** that contains the attributes **STREET** and **CITY**, enter



```

CREATE TYPE ADDRESS AS OBJECT
( STREET VARCHAR2(20),
  CITY   VARCHAR2(20)
);
/

```



Type created.



DESCRIBE address



```

Name          Null?   Type
-----
STREET        VARCHAR2(20)
CITY          VARCHAR2(20)

```

To create and describe the object type **EMPLOYEE** that contains the attributes **LAST_NAME**, **EMPADDR**, **JOB_ID** and **SALARY**, enter



```
CREATE TYPE EMPLOYEE AS OBJECT
(LAST_NAME VARCHAR2(30),
EMPADDR ADDRESS,
JOB_ID VARCHAR2(20),
SALARY NUMBER(7,2)
);
/
```



Type created.



```
DESCRIBE employee
```



Name	Null?	Type
-----	-----	-----
LAST_NAME		VARCHAR2(30)
EMPADDR		ADDRESS
JOB_ID		VARCHAR2(20)
SALARY		NUMBER(7,2)

To create and describe the object type **addr_type** as a table of the object type **ADDRESS**, enter



```
CREATE TYPE addr_type IS TABLE OF ADDRESS;
/
```



Type created.



```
DESCRIBE addr_type
```



addr_type TABLE OF ADDRESS		
Name	Null?	Type
-----	-----	-----
STREET		VARCHAR2(20)
CITY		VARCHAR2(20)

To create and describe the object type **addr_varray** as a varray of the object type **ADDRESS**, enter



```
CREATE TYPE addr_varray AS VARRAY(10) OF ADDRESS;
/
```



Type created.

DESCRIBE



```
DESCRIBE addr_varray
```



```
addr_varray VARRAY(10) OF ADDRESS
Name                                         Null?   Type
-----
STREET                                       VARCHAR2(20)
CITY                                         VARCHAR2(20)
```

To create and describe the table department that contains the columns DEPARTMENT_ID, PERSON and LOC, enter



```
CREATE TABLE department
(DEPARTMENT_ID NUMBER,
PERSON EMPLOYEE,
LOC NUMBER
);
```



```
/
Table created.
```



```
DESCRIBE department
```



```
Name                                         Null?   Type
-----
DEPARTMENT_ID                               NUMBER
PERSON                                       EMPLOYEE
LOC                                           NUMBER
```

To create and describe the object type rational that contains the attributes NUMERATOR and DENOMINATOR, and the METHOD *rational_order*, enter



```
CREATE OR REPLACE TYPE rational AS OBJECT
(NUMERATOR NUMBER,
DENOMINATOR NUMBER,
MAP MEMBER FUNCTION rational_order -
RETURN DOUBLE PRECISION,
PRAGMA RESTRICT_REFERENCES
(rational_order, RNDS, WNDS, RNPS, WNPS) );
/
CREATE OR REPLACE TYPE BODY rational AS OBJECT
MAP MEMBER FUNCTION rational_order -
RETURN DOUBLE PRECISION IS
BEGIN
    RETURN NUMERATOR/DENOMINATOR;
END;
END;
/
```




```
DESCRIBE rational
```



```

Name                                Null?   Type
-----
NUMERATOR                            NUMBER
DENOMINATOR                          NUMBER

METHOD
-----
MAP MEMBER FUNCTION RATIONAL_ORDER RETURNS NUMBER
```

To format the DESCRIBE output use the SET command as follows:



```

SET LINESIZE 80
SET DESCRIBE DEPTH 2
SET DESCRIBE INDENT ON
SET DESCRIBE LINE OFF
```

To display the settings for the object, use the SHOW command as follows:



```
SHOW DESCRIBE
```



```
describe DEPTH 2 LINENUM OFF INDENT ON
```



```
DESCRIBE employee
```



```

Name                                Null?   Type
-----
FIRST_NAME                          VARCHAR2(30)
EMPADDR                              ADDRESS
  STREET                             VARCHAR2(20)
  CITY                               VARCHAR2(20)
JOB_ID                               VARCHAR2(20)
SALARY                              NUMBER(7,2)
```

For more information on using the CREATE TYPE command, see your *Oracle9i SQL Reference*.

For information about using the SET DESCRIBE and SHOW DESCRIBE commands, see the SET and SHOW commands later in this chapter.

DISCONNECT

Syntax

```
DISC[ONNECT]
```

Commits pending changes to the database and logs the current username out of Oracle, but does not exit SQL*Plus.

Usage

Use DISCONNECT within a command file to prevent user access to the database when you want to log the user out of Oracle but have the user remain in SQL*Plus. Use EXIT or QUIT to log out of Oracle and return control to your host computer's operating system.

Examples

Your command file might begin with a CONNECT command and end with a DISCONNECT, as shown below.



```
CONNECT HR/HR
SELECT LAST_NAME, DEPARTMENT_NAME FROM EMP_DETAILS_VIEW;
DISCONNECT
SET INSTANCE FIN2
CONNECT HR2/HR2
```

EXECUTE

Syntax

EXEC[UTE] *statement*

Executes a single PL/SQL statement. The EXECUTE command is often useful when you want to execute a PL/SQL statement that references a stored procedure. For more information on PL/SQL, see your *PL/SQL User's Guide and Reference*.

Terms

Refer to the following for a description of the term or clause:

statement

Represents a PL/SQL statement.

Usage

If your EXECUTE command cannot fit on one line because of the PL/SQL statement, use the SQL*Plus continuation character (a hyphen).

The length of the command and the PL/SQL statement cannot exceed the length defined by SET LINESIZE.

Examples



If the variable :n has been defined with:

```
VARIABLE n NUMBER
```



The following EXECUTE command assigns a value to the bind variable n:

```
EXECUTE :n := 1
```



PL/SQL procedure successfully completed.

For information on how to create a bind variable, see the VARIABLE command in this chapter.

HELP

Syntax

HELP [*topic*]

Accesses the SQL*Plus help system. Enter HELP INDEX for a list of topics.

Terms

Refer to the following for a description of the term or clause:

topic

Represents a SQL*Plus help topic, for example, COLUMN.

Enter HELP without *topic* to get help on the help system.

Usage

You can only enter one topic after HELP. You can abbreviate the topic (for example, COL for COLUMN). However, if you enter only an abbreviated topic and the abbreviation is ambiguous, SQL*Plus displays help for all topics that match the abbreviation. For example, if you enter

```
HELP EX
```

SQL*Plus displays the syntax for the EXECUTE command followed by the syntax for the EXIT command.

If you get a response indicating that help is not available, consult your database administrator.

Examples



To see a list of SQL*Plus commands for which help is available, enter

```
HELP INDEX
```

Alternatively, to see a single column display of SQL*Plus commands for which help is available, enter



```
HELP TOPICS
```

LIST

Syntax

```
L[IST] [n|n m|n*|n LAST|* n|* LAST|LAST]
```

Lists one or more lines of the SQL buffer.

Terms

Refer to the following list for a description of each term or clause:

<i>n</i>	Lists line <i>n</i> .
<i>n m</i>	Lists lines <i>n</i> through <i>m</i> .
<i>n*</i>	Lists line <i>n</i> through the current line.
<i>n LAST</i>	Lists line <i>n</i> through the last line.
<i>*</i>	Lists the current line.
<i>* n</i>	Lists the current line through line <i>n</i> .
<i>* LAST</i>	Lists the current line through the last line.
<i>LAST</i>	Lists the last line.

Enter LIST with no clauses to list all lines. The last line or only line listed becomes the new current line (marked by an asterisk).

Examples



To list the contents of the buffer, enter

```
LIST
```

You will see a listing of all lines in the buffer, similar to the following example:



```
1 SELECT LAST_NAME, DEPARTMENT_ID, JOB_ID
2 FROM EMP_DETAILS_VIEW
3 WHERE JOB_ID = 'SH_CLERK'
4* ORDER BY DEPARTMENT_ID
```

The asterisk indicates that line 4 is the current line.

To list the second line only, enter



```
LIST 2
```

The second line is displayed:



```
2* FROM EMP_DETAILS_VIEW
```

To list the current line (now line 2) to the last line, enter



```
LIST * LAST
```

You will then see this:



```
2 FROM EMP_DETAILS_VIEW
3 WHERE JOB_ID = 'SH_CLERK'
4* ORDER BY DEPARTMENT_ID
```

PRINT

Syntax

PR[INT] [*variable ...*]

Displays the current value of bind variables. For more information on bind variables, see your *PL/SQL User's Guide and Reference*.

Terms

Refer to the following for a description of the clause or term:

variable ...

Represents the names of the bind variables whose values you wish to display.

Enter PRINT with no variables to print all bind variables.

Usage

Bind variables are created using the VARIABLE command. For more information and examples, see the VARIABLE command in this chapter.

You can control the formatting of the PRINT output just as you would query output. For more information.

To automatically display bind variables referenced in a successful PL/SQL block or used in an EXECUTE command, use the AUTOPRINT clause of the SET command. For more information, see the SET command in this chapter.

Examples



The following example illustrates a PRINT command:

```
VARIABLE n NUMBER
BEGIN
:n := 1;
END;
/
PL/SQL procedure successfully completed.
```





PRINT n

N

1

PROMPT

Syntax

```
PRO[MPT] [text]
```

Sends the specified message or a blank line to the user's screen.

Terms

Refer to the following for a description of the term or clause:

text

Represents the text of the message you wish to display. If you omit *text*, PROMPT displays a blank line on the user's screen.

Usage

You can use this command in command files to give information to the user.

Examples

The following example shows the use of PROMPT in conjunction with ACCEPT in a command file called ASKFORDEPT. ASKFORDEPT contains the following SQL*Plus and SQL commands:



```
PROMPT
PROMPT Please enter a valid department
PROMPT For example: 10, 20, 30, 40
ACCEPT NEWDEPT NUMBER PROMPT 'Department ID?> '
SELECT DEPARTMENT_NAME FROM EMP_DETAILS_VIEW
WHERE DEPARTMENT_ID = &NEWDEPT
```

Assume you run the file using START or @:



```
@ASKFORDEPT
```



```
Please enter a valid department
For example: 10, 20, 30, 40
Department ID?>
```

You can enter a department number at the prompt Department ID?>. By default, SQL*Plus lists the line containing &NEWDEPT before and after substitution, and then displays the department name corresponding to the number entered at the Department ID?> prompt. You can use SET VERIFY OFF to prevent this behavior.

RECOVER

Syntax

```
RECOVER {general | managed | END BACKUP}
```

where the *general* clause has the following syntax:

```
[AUTOMATIC] [FROM location]
{ {full_database_recovery | partial_database_recovery | LOGFILE filename}
  [ {TEST | ALLOW integer CORRUPTION } [TEST | ALLOW integer CORRUPTION ]... ]
  [CONTINUE [DEFAULT]] [CANCEL]}
```

where the *full_database_recovery* clause has the following syntax:

```
[STANDBY] DATABASE
  [ {UNTIL {CANCEL | TIME date | CHANGE integer} | USING BACKUP CONTROLFILE}
    [UNTIL {CANCEL | TIME date | CHANGE integer} | USING BACKUP CONTROLFILE]...]
```

where the *partial_database_recovery* clause has the following syntax:

```
{TABLESPACE tablespace [, tablespace]... | DATAFILE datafilename [, datafilename]...
  / STANDBY
  {TABLESPACE tablespace [, tablespace]... | DATAFILE datafilename [, datafilename]...}
  UNTIL [CONSISTENT] [WITH] CONTROLFILE }
```

where the *managed* clause has the following syntax:

```
MANAGED STANDBY DATABASE
  [ {NODELAY | [TIMEOUT] integer | CANCEL [IMMEDIATE] [NOWAIT]}
    | [DISCONNECT [FROM SESSION] ] [FINISH [NOWAIT] ] ]
```

Performs media recovery on one or more tablespaces, one or more datafiles, or the entire database. For more information on the RECOVER command, see the *Oracle9i Administrator's Guide*, the ALTER DATABASE RECOVER command in the *Oracle9i SQL Reference*, the *Oracle9i Backup and Recovery Concepts* guide, and the *Oracle9i User-Managed Backup and Recovery Guide*.

As iSQL*Plus cannot handle interactive commands, you must set AUTORECOVERY ON to use the RECOVER command. Attempting to RECOVER a database with AUTORECOVERY OFF raises the error:

```
SP2-0872 SET AUTORECOVERY ON must be used in iSQL*Plus
```

Terms

Refer to the following list for a description of each term and clause:

AUTOMATIC

Automatically generates the name of the next archived redo log file needed to continue the recovery operation. Oracle uses the LOG_ARCHIVE_DEST (or LOG_ARCHIVE_DEST_1) and LOG_ARCHIVE_FORMAT parameters (or their defaults) to generate the target redo log filename. If the file is found, the redo contained in that file is applied. If the file is not found, SQL*Plus prompts you for a filename, displaying a generated filename as a suggestion.

If you do not specify either AUTOMATIC or LOGFILE, SQL*Plus prompts you for a filename, suggesting the generated filename. You can then accept the generated filename or replace it with a fully qualified filename. If you know the archived filename differs from what Oracle would generate, you can save time by using the LOGFILE clause.

FROM *location*

Specifies the location from which the archived redo log file group is read. The value of location must be a fully specified file location following the conventions of your operating system. If you omit this parameter, SQL*Plus assumes the archived redo log file group is in the location specified by the initialization parameter LOG_ARCHIVE_DEST or LOG_ARCHIVE_DEST_1. Do not specify FROM if you have set a file with SET LOGSOURCE.

LOGFILE

Continues media recovery by applying the specified redo log file. In interactive recovery mode (AUTORECOVERY OFF), if a bad log name is entered, errors for the bad log name are displayed and you are prompted to enter a new log name.

TEST ALLOW integer CORRUPTION

In the event of logfile corruption, specifies the number of corrupt blocks that can be tolerated while allowing recovery to proceed. During normal recovery, integer cannot exceed 1.

CONTINUE

Continues multi-instance recovery after it has been interrupted to disable a thread.

CONTINUE DEFAULT

Continues recovery using the redo log file generated automatically by Oracle if no other logfile is specified. This is equivalent to specifying **AUTOMATIC**, except that Oracle does not prompt for a filename.

CANCEL

Terminates cancel-based recovery.

STANDBY DATABASE

Recovers the standby database using the control file and archived redo log files copied from the primary database. The standby database must be mounted but not open.

DATABASE

Recovers the entire database.

UNTIL CANCEL

Specifies an incomplete, cancel-based recovery. Recovery proceeds by prompting you with suggested filenames of archived redo log files, and recovery completes when you specify **CANCEL** instead of a filename.

UNTIL TIME

Specifies an incomplete, time-based recovery. Use single quotes, and the following format:

```
'YYYY-MM-DD:HH24:MI:SS'
```

UNTIL CHANGE

Specifies an incomplete, change-based recovery. *integer* is the number of the System Change Number (SCN) following the last change you wish to recover. For example, if you want to restore your database up to the transaction with an SCN of 9, you would specify **UNTIL CHANGE 10**.

USING BACKUP CONTROLFILE

Specifies that a backup of the control file be used instead of the current control file.

TABLESPACE

Recovers a particular tablespace. *tablespace* is the name of a tablespace in the current database. You may recover up to 16 tablespaces in one statement.

DATAFILE

Recovers a particular datafile. You can specify any number of datafiles.

STANDBY TABLESPACE

Reconstructs a lost or damaged tablespace in the standby database using archived redo log files copied from the primary database and a control file.

STANDBY DATAFILE

Reconstructs a lost or damaged datafile in the standby database using archived redo log files copied from the primary database and a control file.

UNTIL CONSISTENT WITH CONTROLFILE

Specifies that the recovery of an old standby datafile or tablespace uses the current standby database control file.

MANAGED STANDBY DATABASE

Specifies sustained standby recovery mode. This mode assumes that the standby database is an active component of an overall standby database architecture. A primary database actively archives its redo log files to the standby site. As these archived redo logs arrive at the standby site, they become available for use by a managed standby recovery operation. Sustained standby recovery is restricted to media recovery.

For more information on the parameters of this clause, see the *Oracle9i User-Managed Backup and Recovery Guide*.

NODELAY

Applies a delayed archive log immediately to the standby database overriding any DELAY setting in the LOG_ARCHIVE_DEST_n parameter on the primary database. If you omit this clause, application of the archive log is delayed according to the parameter setting. If DELAY was not specified in the parameter, the archive log is applied immediately.

TIMEOUT

Specifies in minutes the wait period of the sustained recovery operation. The recovery process waits for integer minutes for a requested archived log redo to be available for writing to the standby database. If the redo log file does not become available within that time, the recov-

ery process terminates with an error message. You can then issue the statement again to return to sustained standby recovery mode.

If you do not specify this clause, the database remains in sustained standby recovery mode until you reissue the statement with the RECOVER CANCEL clause or until instance shutdown or failure.

CANCEL (managed clause)

In managed recovery, CANCEL terminates the managed standby recovery operation after applying the current archived redo file. Session control returns when the recovery process terminates.

CANCEL IMMEDIATE

Terminates the managed recovery operation after applying the current archived redo file or after the next redo log file read, whichever comes first. Session control returns when the recovery process terminates.

CANCEL IMMEDIATE cannot be issued from the same session that issued the RECOVER MANAGED STANDBY DATABASE statement.

CANCEL NOWAIT

Terminates the managed recovery operation after the next redo log file read and returns session control immediately.

DISCONNECT FROM SESSION

Indicates that the managed redo process (MRP) should apply archived redo files as a detached background process. Doing so leaves the current session available.

FINISH

Recovers the current log standby logfiles of the standby database. It is useful in the event of the primary database failure. This clause overrides any delays specified for archivelogs, so that Oracle logs are applied immediately.

NOWAIT

Returns control immediately rather than after completion of the recovery process.

Usage

You must have the OSDBA role enabled. You cannot use the RECOVER command when connected via the multi-threaded server.

To perform media recovery on an entire database (all tablespaces), the database must be mounted and closed, and all tablespaces requiring recovery must be online.

To perform media recovery on a tablespace, the database must be mounted and open, and the tablespace must be offline.

To perform media recovery on a datafile, the database can remain open and mounted with the damaged datafiles offline (unless the file is part of the SYSTEM tablespace).

Before using the RECOVER command you must have restored copies of the damaged datafile(s) from a previous backup. Be sure you can access all archived and online redo log files dating back to when that backup was made.

When another log file is required during recovery, a prompt suggests the names of files that are needed. The name is derived from the values specified in the initialization parameters LOG_ARCHIVE_DEST and LOG_ARCHIVE_FORMAT. You should restore copies of the archived redo log files needed for recovery to the destination specified in LOG_ARCHIVE_DEST, if necessary. You can override the initialization parameters by setting the LOGSOURCE variable with the SET LOGSOURCE command.

During recovery you can accept the suggested log name by pressing return, cancel recovery by entering CANCEL instead of a log name, or enter AUTO at the prompt for automatic file selection without further prompting.

If you have enabled autorecovery (that is, SET AUTORECOVERY ON), recovery proceeds without prompting you with filenames. Status messages are displayed when each log file is applied. When normal media recovery is done, a completion status is returned.

Examples



To recover the entire database, enter

```
RECOVER DATABASE
```



To recover the database until a specified time, enter

```
RECOVER DATABASE UNTIL TIME 01-JAN-2001:04:32:00
```

To recover the two tablespaces ts_one and ts_two from the database, enter

RECOVER



```
RECOVER TABLESPACE ts_one, ts_two
```

To recover the datafile data1.db from the database, enter



```
RECOVER DATAFILE 'data1.db'
```

REMARK

Syntax

```
REM[ARK]
```

Begins a comment in a command file. SQL*Plus does not interpret the comment as a command.

Usage

The REMARK command must appear at the beginning of a line, and the comment ends at the end of the line. A line cannot contain both a comment and a command.

Examples



The following command file contains some typical comments:

```
REM COMPUTE uses BREAK ON REPORT to break on end of table
BREAK ON REPORT
COMPUTE SUM OF "DEPARTMENT 10" "DEPARTMENT 20" -
"DEPARTMENT 30" "TOTAL BY JOB_ID" ON REPORT
REM Each column displays the sums of salaries by job for
REM one of the departments 10, 20, 30.
SELECT JOB_ID,
SUM(DECODE( DEPARTMENT_ID, 10, SALARY, 0)) "DEPARTMENT 10",
SUM(DECODE( DEPARTMENT_ID, 20, SALARY, 0)) "DEPARTMENT 20",
SUM(DECODE( DEPARTMENT_ID, 30, SALARY, 0)) "DEPARTMENT 30",
SUM(SALARY) "TOTAL BY JOB_ID"
FROM EMP_DETAILS_VIEW
GROUP BY JOB_ID;
```

REPFOOTER

Syntax

```
REPFOOTER [PAGE] [printspec [text|variable] ...] | [ON|OFF]
```

Places and formats a specified report footer at the bottom of each report, or lists the current REPFOOTER definition.

Terms

Refer to the REPHEADER command for additional information on terms and clauses in the REPFOOTER command syntax.

Enter REPFOOTER with no clauses to list the current REPFOOTER definition.

Usage

If you do not enter a *printspec* clause before the text or variables, REPFOOTER left justifies the text or variables.

You can use any number of constants and variables in a *printspec*. SQL*Plus displays the constants and variables in the order you specify them, positioning and formatting each constant or variable as specified by the *printspec* clauses that precede it.

Note: If SET EMBEDDED is ON, the report footer is suppressed.

Examples

To define “END EMPLOYEE LISTING REPORT” as a report footer on a separate page and to center it, enter:



```
REPFOOTER PAGE CENTER 'END EMPLOYEE LISTING REPORT'
TTITLE RIGHT 'Page: ' FORMAT 999 SQL.PNO
SELECT LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000;
```



LAST_NAME	SALARY
-----	-----
King	24000
Kochhar	17000
De Haan	17000

Russell	14000
Partners	13500
Hartstein	13000

sum	98500

Page: 2

END EMPLOYEE LISTING REPORT

6 rows selected.

To suppress the report footer without changing its definition, enter



REPFOOTER OFF

You can stop the REPHEADER display by setting it off with:



REPHEADER OFF

REPHEADER

Syntax

REPH[EADER] [PAGE] [*printspec* [*text*{*variable*} ...] | [ON|OFF]

where *printspec* represents one or more of the following clauses used to place and format the *text*:

COL *n*
S[KIP] [*n*]
TAB *n*
LE[FT]
CE[NTER]
R[IGHT]
BOLD
FORMAT *text*

Places and formats a specified report header at the top of each report, or lists the current REPHEADER definition.

Terms

Refer to the following list for a description of each term or clause. These terms and clauses also apply to the REPFOOTER command.

PAGE

Begins a new page after printing the specified report header or before printing the specified report footer.

text

Represents the report header or footer text. Enter *text* in single quotes if you want to place more than one word on a single line. The default is NULL.

variable

Represents a user variable or any of the following system-maintained values. SQL.LNO is the current line number, SQL.PNO is the current page number, SQL.RELEASE is the current Oracle release number, SQL.CODE is the current error code, and SQL.USER is the current user-name.

To print one of these values, reference the appropriate variable in the report header or footer. You can format *variable* with the FORMAT clause.

OFF

Turns the report header or footer off (suppresses its display) without affecting its definition.

COL *n*

Indents to column *n* of the current line (backward if column *n* has been passed). Column in this context means print position, not table column.

S[KIP] [*n*]

Skips to the start of a new line *n* times; if you omit *n*, one time; if you enter zero for *n*, backward to the start of the current line.

TAB *n*

Skips forward *n* columns (backward if you enter a negative value for *n*). Column in this context means print position, not table column.

LE[FT] CE[NTER] R[IGHT]

Left-align, center, and right-align data on the current line respectively. SQL*Plus aligns following data items as a group, up to the end of the *printspect* or the next LEFT, CENTER, RIGHT, or COL command. CENTER and RIGHT use the SET LINESIZE value to calculate the position of the data item that follows.

BOLD

Prints data in bold print. SQL*Plus represents bold print on your terminal by repeating the data on three consecutive lines. On some operating systems, SQL*Plus may instruct your printer to print bold text on three consecutive lines, instead of bold.

FORMAT *text*

Specifies a format model that determines the format of following data items, up to the next FORMAT clause or the end of the command. The format model must be a *text* constant such as A10 or \$999.

If the datatype of the format model does not match the datatype of a given data item, the FORMAT clause has no effect on that item.

If no appropriate `FORMAT` model precedes a given data item, `SQL*Plus` prints `NUMBER` values according to the format specified by `SET NUMFORMAT` or, if you have not used `SET NUMFORMAT`, the default format. `SQL*Plus` prints `DATE` values using the default format.

Enter `REPHEADER` with no clauses to list the current `REPHEADER` definition.

Usage

If you do not enter a *printspec* clause before the text or variables, `REPHEADER` left justifies the text or variables.

You can use any number of constants and variables in a *printspec*. `SQL*Plus` displays the constants and variables in the order you specify, positioning and formatting each constant or variable as specified by the *printspec* clauses that precede it.

Examples

To define “EMPLOYEE LISTING REPORT” as a report header on a separate page, and to center it, enter:



```
REPHEADER PAGE CENTER 'EMPLOYEE LISTING REPORT'
TTITLE RIGHT 'Page: ' FORMAT 999 SQL.PNO
SELECT LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE SALARY > 12000;
```



```

Page:      1
                EMPLOYEE LISTING REPORT

Page:      2

LAST_NAME          SALARY
-----
King                24000
Kochhar             17000
De Haan             17000
Russell             14000
Partners            13500
Hartstein           13000
-----
sum                 98500

6 rows selected.
```

RUN

Syntax

R[UN]

Lists and executes the SQL command or PL/SQL block currently stored in the SQL buffer.

Usage

RUN causes the last line of the SQL buffer to become the current line.

The slash command (/) functions similarly to RUN, but does not list the command in the SQL buffer on your screen.

Examples



Assume the SQL buffer contains the following query:

```
SELECT DEPARTMENT_ID
FROM EMP_DETAILS_VIEW
WHERE SALARY>12000
```



To RUN the query, enter

```
RUN
```



```
1 SELECT DEPARTMENT_ID
2 FROM EMP_DETAILS_VIEW
3* WHERE SALARY>12000
```

```
DEPARTMENT_ID
```

```
-----
          90
          90
          90
          80
          80
          20
```

```
6 rows selected.
```

SET

Syntax

SET *system_variable* *value*

where *system_variable* and *value* represent one of the following clauses.

```
APPI[NFO]{ON|OFF|text}
ARRAY[SIZE] {15|n}
AUTO[COMMIT]{ON|OFF|IMM[EDIATE]|n}
AUTOP[RINT] {ON|OFF}
AUTORECOVERY {ON|OFF}
AUTOT[RACE] {ON|OFF|TRACE[ONLY]} [EXP[LAIN]] [STAT[ISTICS]]
BLO[CKTERMINATOR] {.|c}
CMDS[EP] {.|c|ON|OFF}
COM[PATIBILITY]{V7|V8|NATIVE}
CON[CAT] {.|c|ON|OFF}
COPYC[OMMIT] {0|n}
COPYTYPECHECK {ON|OFF}
DEF[INE] {&|c|ON|OFF}
DESCRIBE [DEPTH {1|n|ALL}][LINENUM {ON|OFF}][INDENT {ON|OFF}]
ECHO {ON|OFF}
EMB[EDDED] {ON|OFF}
ESC[APE] {\|c|ON|OFF}
FEED[BACK] {6|n|ON|OFF}
FLAGGER {OFF|ENTRY |INTERMED[IATE]|FULL}
HEA[DING] {ON|OFF}
HEADS[EP] {||c|ON|OFF}
INSTANCE [instance_path|LOCAL]
LIN[ESIZE] {80|n}
LOBOF[FSET] {n|1}
LOGSOURCE [pathname]
LONG {80|n}
LONGC[HUNKSIZE] {80|n}
MARK[UP] HTML [HEAD text] [BODY text] [TABLE text] [ENTMAP {ON|OFF}] [PRE[FORMAT]
{ON|OFF}]
NULL text
NUMF[ORMAT] format
NUM[WIDTH] {10|n}
PAGES[IZE] {24|n}
RECSEP {WR[APPED]|EA[CH]|OFF}
```



```

RECSEPCHAR {[_]c}
SERVEROUT[PUT] {ON|OFF} [SIZE n] [FOR[MAT] {WRA[PPED]|WOR[D_
WRAPPED]|TRU[NCATED]]}
SQLC[ASE] {MIX[ED]|LO[WER]|UP[PER]}
SQLPLUSCOMPAT[IBILITY] {x.y[z]}
SQLT[ERMINATOR] {[:|c|ON|OFF]}
TIMI[NG] {ON|OFF}
UND[ERLINE] {-|c|ON|OFF}
VER[IFY] {ON|OFF}
WRA[P] {ON|OFF}

```

Sets a system variable to alter the SQL*Plus environment settings for your current session, for example:

- the display width for NUMBER data
- turning on HTML formatting
- enabling or disabling the printing of column headings
- the number of lines per page

Terms

Refer to the following list for a description of each term, clause, or system variable:

APP[INFO]{ON|OFF} *text*

Sets automatic registering of command files through the DBMS_APPLICATION_INFO package. This enables the performance and resource usage of each command file to be monitored by your DBA. The registered name appears in the MODULE column of the V\$SESSION and V\$SQLAREA virtual tables. You can also read the registered name using the DBMS_APPLICATION_INFO.READ_MODULE procedure.

ON registers command files invoked by the @, @@ or START commands. OFF disables registering of command files. Instead, the current value of *text* is registered. *Text* specifies the text to register when no command file is being run or when APPINFO is OFF, which is the default. The default for *text* is "SQL*Plus". If you enter multiple words for *text*, you must enclose them in quotes. The maximum length for *text* is limited by the DBMS_APPLICATION_INFO package.

The registered name has the format *nn@xfilename* where: *nn* is the depth level of command file; *x* is '<' when the command file name is trun-

cated, otherwise, it is blank; and *filename* is the command file name, possibly truncated to the length allowed by the DBMS_APPLICATION_INFO package interface.

Note: To use this feature, you must have access to the DBMS_APPLICATION_INFO package. Run DBMSUTIL.SQL (this name may vary depending on your operating system) as SYS to create the DBMS_APPLICATION_INFO package. DBMSUTIL.SQL is part of the Oracle9i database server product.

For more information on the DBMS_APPLICATION_INFO package, see the *Oracle9i Performance Guide and Reference* manual.

ARRAY[SIZE] {15|*n*}

Sets the number of rows—called a *batch*—that SQL*Plus will fetch from the database at one time. Valid values are 1 to 5000. A large value increases the efficiency of queries and subqueries that fetch many rows, but requires more memory. Values over approximately 100 provide little added performance. ARRAYSIZE has no effect on the results of SQL*Plus operations other than increasing efficiency.

AUTO[COMMIT]{ON|OFF|IMMEDIATE}|*n*}

Controls when Oracle commits pending changes to the database. ON commits pending changes to the database after Oracle executes each successful INSERT, UPDATE, or DELETE command or PL/SQL block. OFF suppresses automatic committing so that you must commit changes manually (for example, with the SQL command COMMIT). IMMEDIATE functions in the same manner as the ON option. *n* commits pending changes to the database after Oracle executes *n* successful SQL INSERT, UPDATE, or DELETE commands or PL/SQL blocks. *n* cannot be less than zero or greater than 2,000,000,000. The statement counter is reset to zero after successful completion of *n* INSERT, UPDATE or DELETE commands or PL/SQL blocks, a commit, a rollback, or a SET AUTOCOMMIT command.

Note: For this feature, a PL/SQL block is considered one transaction, regardless of the actual number of SQL commands contained within it.

AUTOP[RINT] {ON|OFF}

Sets the automatic PRINTing of bind variables. ON or OFF controls whether SQL*Plus automatically displays bind variables (referenced in a successful PL/SQL block or used in an EXECUTE command). For more information about displaying bind variables, see the PRINT command in this chapter.

AUTORECOVERY [ON|OFF]

ON sets the RECOVER command to automatically apply the default filenames of archived redo log files needed during recovery. No interaction is needed when AUTORECOVERY is set to ON, provided the necessary files are in the expected locations with the expected names. The filenames used when AUTORECOVERY is ON are derived from the values of the initialization parameters LOG_ARCHIVE_DEST and LOG_ARCHIVE_FORMAT.

As iSQL*Plus cannot handle interactive commands, you must set AUTORECOVERY ON to use the RECOVER command. Attempting to RECOVER a database with AUTORECOVERY OFF raises the error:

```
SP2-0872 SET AUTORECOVERY ON must be used in iSQL*Plus
```

AUTOT[RACE] {ON|OFF|TRACE[ONLY]} [EXP[LAIN]] [STAT[ISTICS]]

Displays a report on the execution of successful SQL DML statements (SELECT, INSERT, UPDATE or DELETE). The report can include execution statistics and the query execution path.

OFF does not display a trace report. ON displays a trace report. TRACEONLY displays a trace report, but does not print query data, if any. EXPLAIN shows the query execution path by performing an EXPLAIN PLAN. STATISTICS displays SQL statement statistics. Information about EXPLAIN PLAN is documented in the *Oracle9i SQL Reference manual*.

Using ON or TRACEONLY with no explicit options defaults to EXPLAIN STATISTICS.

The TRACEONLY option may be useful to suppress the query data of large queries. If STATISTICS is specified, SQL*Plus still fetches the query data from the server, however, the data is not displayed.

The AUTOTRACE report is printed after the statement has successfully completed.

Information about Execution Plans and the statistics is documented in the *Oracle9i Performance Guide and Reference* manual.

When SQL*Plus produces a STATISTICS report, a second connection to the database is automatically created. This connection is closed when the STATISTICS option is set to OFF, or you log out of SQL*Plus.

The formatting of your AUTOTRACE report may vary depending on the version of the server to which you are connected and the configuration of the server.

AUTOTRACE is not available when FIPS flagging is enabled.

BLOCKTERMINATOR [*c*]

Sets the non-alphanumeric character used to end PL/SQL blocks to *c*. It cannot be an alphanumeric character or a whitespace. To execute the block, you must issue a RUN or / (slash) command.

CMDS[EP] {*c*|ON|OFF}

Sets the non-alphanumeric character used to separate multiple SQL*Plus commands entered on one line to *c*. ON or OFF controls whether you can enter multiple commands on a line. ON automatically sets the command separator character to a semicolon (;).

COMPATIBILITY]{V7|V8|NATIVE}

Specifies the version of Oracle to which you are currently connected. Set COMPATIBILITY to V7 for Oracle7, or to V8 for Oracle8 or later. Set COMPATIBILITY to NATIVE if you wish the database to determine the setting (If connected to Oracle9i, compatibility defaults to NATIVE). COMPATIBILITY should be correctly set for the version of Oracle to which you are connected; otherwise, you may be unable to run any SQL commands.

Note: You can set COMPATIBILITY to V7 or V8 when connected to Oracle9i. This enables you to run Oracle7 SQL, Oracle8 or Oracle8i SQL against Oracle9i.

CON[CAT] [*c*|ON|OFF}

Sets the character you can use to terminate a substitution variable reference if you wish to immediately follow the variable with a character that SQL*Plus would otherwise interpret as a part of the substitution

variable name. SQL*Plus resets the value of CONCAT to a period when you switch CONCAT on.

COPYC[OMMIT] {0|n}

Controls the number of batches after which the COPY command commits changes to the database. COPY commits rows to the destination database each time it copies *n* row batches. Valid values are zero to 5000. You can set the size of a batch with the ARRAYSIZE variable. If you set COPYCOMMIT to zero, COPY performs a commit only at the end of a copy operation.

COPYTYPECHECK {ON|OFF}

Sets the suppression of the comparison of datatypes while inserting or appending to tables with the COPY command. This is to facilitate copying to DB2, which requires that a CHAR be copied to a DB2 DATE.

DEFINE {&|c|ON|OFF}

Sets the character used to prefix substitution variables to *c*. ON or OFF controls whether SQL*Plus will scan commands for substitution variables and replace them with their values. ON changes the value of *c* back to the default '&', not the most recently used character. The setting of DEFINE to OFF overrides the setting of the SCAN variable.

DESCRIBE [DEPTH {1|n|ALL}][LINENUM {ON|OFF}][INDENT {ON|OFF}]

Sets the depth of the level to which you can recursively describe an object. The valid range of the DEPTH clause is from 1 to 50. If you SET DESCRIBE DEPTH ALL, then the depth will be set to 50, which is the maximum level allowed. You can also display the line number and indentation of the attribute or column name when an object contains multiple object types. Use the SET LINESIZE command to control the width of the data displayed.

For more information about describing objects, see DESCRIBE earlier in this chapter.

ECHO {ON|OFF}

Controls whether the START command lists each command in a command file as the command is executed. ON lists the commands; OFF suppresses the listing.

EMB[EDDED] {ON|OFF}

Controls where on a page each report begins. OFF forces each report to start at the top of a new page. ON allows a report to begin anywhere on a page. Set EMBEDDED to ON when you want a report to begin printing immediately following the end of the previously run report.

ESC[APE] {\|c|ON|OFF}

Defines the character you enter as the escape character. OFF undefines the escape character. ON enables the escape character. ON changes the value of *c* back to the default “\”.

You can use the escape character before the substitution character (set through SET DEFINE) to indicate that SQL*Plus should treat the substitution character as an ordinary character rather than as a request for variable substitution.

FEED[BACK] {6|*n*|ON|OFF}

Displays the number of records returned by a query when a query selects at least *n* records. ON or OFF turns this display on or off. Turning feedback ON sets *n* to 1. Setting feedback to zero is equivalent to turning it OFF.

FLAGGER {OFF|ENTRY |INTERMED[IATE]|FULL}

Checks to make sure that SQL statements conform to the ANSI/ISO SQL92 standard. If any non-standard constructs are found, the Oracle Server flags them as errors and displays the violating syntax. This is the equivalent of the SQL language ALTER SESSION SET FLAGGER command.

You may execute SET FLAGGER even if you are not connected to a database. FIPS flagging will remain in effect across SQL*Plus sessions until a SET FLAGGER OFF (or ALTER SESSION SET FLAGGER = OFF) command is successful or you exit SQL*Plus.

When FIPS flagging is enabled, SQL*Plus displays a warning for the CONNECT, DISCONNECT, and ALTER SESSION SET FLAGGER commands, even if they are successful.

HEA[DING] {ON|OFF}

Controls printing of column headings in reports. ON prints column headings in reports; OFF suppresses column headings.

The SET HEADING OFF command will not affect the column width displayed, and only suppresses the printing of the column header itself.

HEADS[EP] {*c*|ON|OFF}

Defines the character you enter as the heading separator character. The heading separator character cannot be alphanumeric or white space. You can use the heading separator character in the COLUMN command and in the old forms of BTITLE and TTITLE to divide a column heading or title onto more than one line. ON or OFF turns heading separation on or off. When heading separation is OFF, SQL*Plus prints a heading separator character like any other character. ON changes the value of *c* back to the default “|”.

INSTANCE [*instance_path*]LOCAL

Changes the default instance for your session to the specified instance path. Using the SET INSTANCE command does not connect to a database. The default instance is used for commands when no instance is specified.

Any commands preceding the first use of SET INSTANCE communicate with the default instance.

To reset the instance to the default value for your operating system, you can either enter SET INSTANCE with no *instance_path* or SET INSTANCE LOCAL. See your operating system-specific Oracle documentation for a description of how to set the initial default instance.

Note, you can only change the instance when you are not currently connected to any instance. That is, you must first make sure that you have disconnected from the current instance, then set or change the instance, and reconnect to an instance in order for the new setting to be enabled.

This command may only be issued when Oracle Net is running. You can use any valid Oracle Net connect identifier as the specified instance path. See your operating system-specific Oracle documentation for a complete description of how your operating system specifies Oracle Net connect identifiers. The maximum length of the instance path is 64 characters.

LIN[ESIZE] {80|*n*}

Sets the total number of characters that SQL*Plus displays on one line before beginning a new line. It also controls the position of centered and right-aligned text in TTITLE, BTITLE, REPHEADER and REP-

FOOTER. You can define LINESIZE as a value from 1 to a maximum that is system dependent. Refer to the Oracle installation and user's manual(s) provided for your operating system.

LOBOF[FSET] {*n*|1}

Sets the starting position from which CLOB and NCLOB data is retrieved and displayed.

LOGSOURCE [*pathname*]

Specifies the location from which archive logs are retrieved during recovery. The default value is set by the LOG_ARCHIVE_DEST initialization parameter in the Oracle initialization file, *init.ora*. Using the SET LOGSOURCE command without a pathname restores the default location.

LONG {80|*n*}

Sets maximum width (in bytes) for displaying LONG, CLOB and NCLOB values; and for copying LONG values. The maximum value of *n* is 2 gigabytes.

LONGC[HUNKSIZE] {80|*n*}

Sets the size (in bytes) of the increments in which SQL*Plus retrieves a LONG, CLOB or NCLOB value.

MARK[UP] HTML [HEAD *text*] [BODY *text*] [TABLE *text*] [ENTMAP {ON|OFF}]
[PRE[FORMAT] {ON|OFF}]

Outputs HTML marked up text, which is the output used by iSQL*Plus. Beware of using options which generate invalid HTML output in iSQL*Plus as it may corrupt the browser screen. The HEAD and BODY options may be useful for dynamic reports and for reports saved to local files.

Use the SHOW MARKUP command to view the status of MARKUP options.

NULL *text*

Sets the text that represents a null value in the result of a SQL SELECT command. Use the NULL clause of the COLUMN command to override the setting of the NULL variable for a given column.

NUMF[ORMAT] *format*

Sets the default format for displaying numbers. Enter a number format for *format*. For number format descriptions, see the FORMAT clause of the COLUMN command in this chapter.

NUM[WIDTH] {10|*n*}

Sets the default width for displaying numbers. For number format descriptions, see the FORMAT clause of the COLUMN command in this chapter.

PAGES[IZE] {24|*n*}

Sets the number of lines in each page. You can set PAGESIZE to zero to suppress all headings, page breaks, titles, the initial blank line, and other formatting information.

RECSEPCHAR { |*c*}

Display or print record separators. A record separator consists of a single line of the RECSEPCHAR (record separating character) repeated LINESIZE times.

RECSEPCHAR defines the record separating character. A single space is the default.

RECSEP {WR[APPED]|EA[CH]|OFF}

RECSEP tells SQL*Plus where to make the record separation. For example, if you set RECSEP to WRAPPED, SQL*Plus prints a record separator only after wrapped lines. If you set RECSEP to EACH, SQL*Plus prints a record separator following every row. If you set RECSEP to OFF, SQL*Plus does not print a record separator.

SERVEROUT[PUT] {ON|OFF} [SIZE *n*] [FOR[MAT] {WR[APPED]|WOR[D]
WRAPPED}|TRU[NCATED]]

Controls whether to display the output (that is, DBMS_OUTPUT.PUT_LINE) of stored procedures or PL/SQL blocks in SQL*Plus. OFF suppresses the output of DBMS_OUTPUT.PUT_LINE; ON displays the output.

SIZE sets the number of bytes of the output that can be buffered within the Oracle8i or Oracle9i database server. The default for *n* is 2000. *n* cannot be less than 2000 or greater than 1,000,000.

When `WRAPPED` is enabled SQL*Plus wraps the server output within the line size specified by `SET LINESIZE`, beginning new lines when required.

When `WORD_WRAPPED` is enabled, each line of server output is wrapped within the line size specified by `SET LINESIZE`. Lines are broken on word boundaries. SQL*Plus left justifies each line, skipping all leading whitespace.

When `TRUNCATED` is enabled, each line of server output is truncated to the line size specified by `SET LINESIZE`.

For each `FORMAT`, every server output line begins on a new output line.

For more information on `DBMS_OUTPUT.PUT_LINE`, see your *Oracle9i Application Developer's Guide - Fundamentals*.

`SQLC[ASE] {MIX[ED]|LO[WER]|UP[PER]}`

Converts the case of SQL commands and PL/SQL blocks just prior to execution. SQL*Plus converts all text within the command, including quoted literals and identifiers, to uppercase if `SQLCASE` equals `UPPER`, to lowercase if `SQLCASE` equals `LOWER`, and makes no changes if `SQLCASE` equals `MIXED`.

`SQLCASE` does not change the SQL buffer itself.

`SQLPLUSCOMPAT[IBILITY] {x.y.z}`

Sets the behavior or output format of `VARIABLE` to that of the release or version specified by `x.y.z`. Where `x` is the *version* number, `y` is the *release* number, and `z` is the *update* number. For example, 8.1, 8.1.7 or 9.0.0. In later releases, `SQLPLUSCOMPATIBILITY` may affect features other than `VARIABLE`.

Setting the value of `SQLPLUSCOMPATIBILITY` to a version less than 9.0.0 will result in `VARIABLE` definition of `NCHAR` or `NVARCHAR2` datatypes to revert to Oracle8i behavior whereby the size of the variable is in bytes or characters depending on the chosen national character set.

The default `glogin.sql` file contains `SET SQLPLUSCOMPAT 8.1.7`. It is recommended that you add `SET SQLPLUSCOMPAT 9.0.0` to your scripts to maximize their compatibility with future versions of SQL*Plus.

`SQLT[ERMINATOR] {c|ON|OFF}`

Sets the character used to end and execute SQL commands to *c*. It cannot be an alphanumeric character or a whitespace. OFF means that SQL*Plus recognizes no command terminator; you terminate a SQL command by entering an empty line. If SQLBLANKLINES is set ON, you must use the BLOCKTERMINATOR to terminate a SQL command. ON resets the terminator to the default semicolon (;).

`TIMI[NG] {ON|OFF}`

Controls the display of timing statistics. ON displays timing statistics on each SQL command or PL/SQL block run. OFF suppresses timing of each command. For information about the data SET TIMING ON displays, see the Oracle installation and user's manual(s) provided for your operating system. Refer to the TIMING command for information on timing multiple commands.

`UND[ERLINE] {c|ON|OFF}`

Sets the character used to underline column headings in SQL*Plus reports to *c*. Note, *c* cannot be an alphanumeric character or a white space. ON or OFF turns underlining on or off. ON changes the value of *c* back to the default “-”.

`VER[IFY] {ON|OFF}`

Controls whether SQL*Plus lists the text of a SQL statement or PL/SQL command before and after SQL*Plus replaces substitution variables with values. ON lists the text; OFF suppresses the listing.

`WRA[P] {ON|OFF}`

Controls whether SQL*Plus truncates the display of a SELECTed row if it is too long for the current line width. OFF truncates the SELECTed row; ON allows the SELECTed row to wrap to the next line.

Use the WRAPPED and TRUNCATED clauses of the COLUMN command to override the setting of WRAP for specific columns.

Usage

SQL*Plus maintains system variables (also called SET command variables) to enable you to setup a particular environment for a SQL*Plus session. You can change these system variables with the SET command and list them with the SHOW command.

SET ROLE and SET TRANSACTION are SQL commands (see the *Oracle9i SQL Reference* for more information). When not followed by the keywords TRANSACTION or ROLE, SET is assumed to be a SQL*Plus command.

Examples

The following examples show sample uses of selected SET command variables.

APPINFO

To display the setting of APPINFO, as it is SET OFF by default, enter



```
SET APPINFO ON
SHOW APPINFO
```



```
APPINFO is ON and set to "SQL*Plus"
```

To change the default text, enter



```
SET APPINFO 'This is SQL*Plus'
```

To make sure that registration has taken place, enter



```
VARIABLE MOD VARCHAR2(50)
VARIABLE ACT VARCHAR2(40)
EXECUTE DBMS_APPLICATION_INFO.READ_MODULE(:MOD, :ACT);
PL/SQL procedure successfully completed.
```



```
PRINT MOD
```



```
MOD
-----
This is SQL*Plus
```

To change APPINFO back to its default setting, enter



```
SET APPI OFF
```

AUTORECOVERY

To set the recovery mode to **AUTOMATIC**, enter



```
SET AUTORECOVERY ON
RECOVER DATABASE
```

CMDSEP

To specify a title with **TTITLE** and format a column with **COLUMN**, both on the same line, enter



```
SET CMDSEP +
TTITLE LEFT 'SALARIES' + COLUMN SALARY FORMAT $99,999
SELECT LAST_NAME, SALARY FROM EMP_DETAILS_VIEW
WHERE JOB_ID = 'SH_CLERK';
```



```
SALARIES
LAST_NAME                SALARY
-----
Taylor                   $3,200
Fleur                    $3,100
Sullivan                  $2,500
Geoni                     $2,800
Sarchand                  $4,200
Bull                      $4,100
Dellinger                 $3,400
Cabrio                    $3,000
Chung                     $3,800
Dilly                     $3,600
Gates                     $2,900
Perkins                   $2,500
Bell                      $4,000
Everett                   $3,900
McCain                    $3,200
Jones                     $2,800
```

```
SALARIES
LAST_NAME                SALARY
-----
Walsh                     $3,100
Feeney                    $3,000
OConnell                  $2,600
Grant                     $2,600
```

20 rows selected.

COMPATIBILITY

To run a command file, `SALARY.SQL`, created with Oracle7 SQL syntax, enter



```
SET COMPATIBILITY V7
START SALARY
```

After running the file, reset compatibility to `NATIVE` to run command files created for Oracle9i:



```
SET COMPATIBILITY NATIVE
```

Alternatively, you can add the command `SET COMPATIBILITY V7` to the beginning of the command file, and reset `COMPATIBILITY` to `NATIVE` at the end of the file.

DESCRIBE

To describe the view `EMP_DETAILS_VIEW` to a depth of two levels, and indent the output while also displaying line numbers, first describe the view as follows:



```
DESCRIBE EMP_DETAILS_VIEW
```



Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
JOB_ID	NOT NULL	VARCHAR2(10)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)
LOCATION_ID		NUMBER(4)
COUNTRY_ID		CHAR(2)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
JOB_TITLE	NOT NULL	VARCHAR2(35)
CITY	NOT NULL	VARCHAR2(30)
STATE_PROVINCE		VARCHAR2(25)
COUNTRY_NAME		VARCHAR2(40)
REGION_NAME		VARCHAR2(25)

To format `EMP_DETAILS_VIEW` so that the output displays with indentation and line numbers, use the `SET DESCRIBE` command as follows:



```
SET DESCRIBE DEPTH 2 LINENUM ON INDENT ON
```

To display the above settings, enter



```
DESCRIBE EMP_DETAILS_VIEW
```



	Name	Null?	Type

1	EMPLOYEE_ID	NOT NULL	NUMBER(6)
2	JOB_ID	NOT NULL	VARCHAR2(10)
3	MANAGER_ID		NUMBER(6)
4	DEPARTMENT_ID		NUMBER(4)
5	LOCATION_ID		NUMBER(4)
6	COUNTRY_ID		CHAR(2)
7	FIRST_NAME		VARCHAR2(20)
8	LAST_NAME	NOT NULL	VARCHAR2(25)
9	SALARY		NUMBER(8,2)
10	COMMISSION_PCT		NUMBER(2,2)
11	DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
12	JOB_TITLE	NOT NULL	VARCHAR2(35)
13	CITY	NOT NULL	VARCHAR2(30)
14	STATE_PROVINCE		VARCHAR2(25)
15	COUNTRY_NAME		VARCHAR2(40)
16	REGION_NAME		VARCHAR2(25)

ESCAPE

If you define the escape character as an exclamation point (!), then



```
SET ESCAPE !
ACCEPT v1 PROMPT 'Enter !&1:'
```

displays this prompt:



```
Enter &1:
```

To set the escape character back to the default value of \ (backslash), enter



```
SET ESCAPE ON
```

HEADING

To suppress the display of column headings in a report, enter



```
SET HEADING OFF
```

If you then run a SQL SELECT command



```
SELECT LAST_NAME, SALARY
FROM EMP_DETAILS_VIEW
WHERE JOB_ID = 'AC_MGR';
```

the following output results:



```
Higgins                12000
```

To turn the display of column headings back on, enter



```
SET HEADING ON
```

INSTANCE

To set the default instance to “PROD1” enter



```
DISCONNECT  
SET INSTANCE PROD1
```

To set the instance back to the default of local, enter



```
SET INSTANCE local
```

You must disconnect from any connected instances to change the instance.

LOBOFFSET

To set the starting position from which a CLOB column’s data is retrieved to the 22nd position, enter



```
SET LOBOFFSET 22
```

The CLOB data will wrap on your screen; SQL*Plus will not truncate until the 23rd character.

LOGSOURCE

To set the default location of log files for recovery to the directory “/usr/oracle90/dbs/arch” enter



```
SET LOGSOURCE "/usr/oracle90/dbs/arch"  
RECOVER DATABASE
```

LONG

To set the maximum number of characters to fetch for displaying and copying LONG values, to 500, enter



```
SET LONG 500
```

The LONG data will wrap on your screen; SQL*Plus will not truncate until the 501st character. The default for LONG is 80 characters.

LONGCHUNKSIZE

To set the size of the increments in which SQL*Plus retrieves LONG values to 100 characters, enter



```
SET LONGCHUNKSIZE 100
```

The LONG data will be retrieved in increments of 100 characters until the entire value is retrieved or the value of SET LONG is reached, whichever is the smaller.

SERVEROUTPUT

To enable the display of text within a PL/SQL block using DBMS_OUTPUT.PUT_LINE, enter



```
SET SERVEROUTPUT ON
```

The following example shows what happens when you execute an anonymous procedure with SET SERVEROUTPUT ON:



```
BEGIN
  DBMS_OUTPUT.PUT_LINE('Task is complete');
END;
/
```



```
Task is complete.
PL/SQL procedure successfully completed.
```

The following example shows what happens when you create a trigger with SET SERVEROUTPUT ON:



```
CREATE TRIGGER SERVER_TRIG BEFORE INSERT OR UPDATE -
OR DELETE
ON SERVER_TAB
BEGIN
  DBMS_OUTPUT.PUT_LINE('Task is complete.');
```



```
END;
/
Trigger created.
```



```
INSERT INTO SERVER_TAB VALUES ('TEXT');
```



```
Task is complete.
1 row created.
```

To set the output to WORD_WRAPPED, enter



```
SET SERVEROUTPUT ON FORMAT WORD_WRAPPED
```

```

SET LINESIZE 20
BEGIN
  DBMS_OUTPUT.PUT_LINE('If there is nothing left to do');
  DBMS_OUTPUT.PUT_LINE('shall we continue with plan B?');
END;
/
If there is nothing
left to do
shall we continue
with plan B?

```



To set the output to TRUNCATED, enter

```

SET SERVEROUTPUT ON FORMAT TRUNCATED
SET LINESIZE 20
BEGIN
  DBMS_OUTPUT.PUT_LINE('If there is nothing left to do');
  DBMS_OUTPUT.PUT_LINE('shall we continue with plan B?');
END;
/
If there is nothing
shall we continue wi

```



SQLCONTINUE

To set the SQL*Plus command continuation prompt to an exclamation point followed by a space, enter

```
SET SQLCONTINUE '! '
```



SQL*Plus will prompt for continuation as follows:

```

TTITLE 'MONTHLY INCOME' -
! RIGHT SQL.PNO SKIP 2 -
! CENTER 'PC DIVISION'

```

The default continuation prompt is ">".

SQLPROMPT

You need the Select Any Table privilege to successfully run the following example scripts.

To set the SQL*Plus command prompt to show your SID, enter

```

SET TERMOUT OFF
COLUMN X NEW_VALUE Y

```



```
SELECT RTRIM(INSTANCE, CHR(0)) X FROM V$THREAD;
SQLPROMPT '&Y SQL>'
SET TERMOUT ON
```

To set the SQL*Plus command prompt to show the current user, enter



```
SET TERMOUT OFF
COLUMN D22 NEW_VALUE VAR
SELECT USERNAME D22 FROM USER_USERS;
SQLPROMPT '&&VAR>'
SET TERMOUT ON
```

These settings are not dynamic. You need to reset them whenever you change instances, such as when you use the connect command to log on to another instance.

SHOW

Syntax

SHO[W] *option*

where *option* represents one of the following terms or clauses:

system_variable

ALL

BTI[TLE]

ERR[ORS] [{ FUNCTION | PROCEDURE | PACKAGE | PACKAGE BODY | TRIGGER
| VIEW | TYPE | TYPE BODY | DIMENSION | JAVA CLASS } [*schema.*]*name*]

LNO

PARAMETERS [*parameter_name*]

PNO

REL[EASE]

REPF[OOTER]

REPH[EADER]

SGA

SPOO[L]

SQLCODE

TTI[TLE]

USER

Shows the value of a SQL*Plus system variable or the current SQL*Plus environment.

Terms

Refer to the following list for a description of each term or clause:

system_variable

Represents any system variable set by the SET command.

ALL

Lists the settings of all SHOW options, except ERRORS and SGA, in alphabetical order.

BTI[TLE]

Shows the current BTITLE definition.

ERR[ORS] [(FUNCTION|PROCEDURE|PACKAGE|PACKAGE BODY|TRIGGER
|VIEW|TYPE|TYPE BODY | DIMENSION | JAVA CLASS) [*schema*.]*name*]

Shows the compilation errors of a stored procedure (includes stored functions, procedures, and packages). After you use the CREATE command to create a stored procedure, a message is displayed if the stored procedure has any compilation errors. To see the errors, you use SHOW ERRORS.

When you specify SHOW ERRORS with no arguments, SQL*Plus shows compilation errors for the most recently created or altered stored procedure. When you specify the type (function, procedure, package, package body, trigger, view, type, type body, dimension, or java class) and the name of the PL/SQL stored procedure, SQL*Plus shows errors for that stored procedure. For more information on compilation errors, see your *PL/SQL User's Guide and Reference*.

schema contains the named object. If you omit *schema*, SHOW ERRORS assumes the object is located in your current schema.

SHOW ERRORS output displays the line and column number of the error (LINE/COL) as well as the error itself (ERROR). LINE/COL and ERROR have default widths of 8 and 65, respectively. You can alter these widths using the COLUMN command.

LNO

Shows the current line number (the position in the current page of the display and/or spooled output).

PARAMETERS [*parameter_name*]

Displays the current values for one or more initialization parameters. You can use a string after the command to see a subset of parameters whose names include that string. For example, if you enter:



```
SHOW PARAMETERS COUNT
```

NAME	TYPE	VALUE
-----	----	-----
db_file_multiblock_read_count	integer	12
spin_count	integer	0

The SHOW PARAMETERS command, without any string following the command, displays all initialization parameters.

Your output may vary depending on the version and configuration of the Oracle database server to which you are connected. You need `SELECT ON V_$PARAMETER` object privileges to use the `PARAMETERS` clause, otherwise you will receive a message

ORA-00942: table or view does not exist

PNO

Shows the current page number.

REL[EASE]

Shows the release number of Oracle that SQL*Plus is accessing.

REPF[OOTER]

Shows the current REPFOOTER definition.

REPH[EADER]

Shows the current REPHEADER definition.

SPOO[L]

Shows whether output is being spooled.

SGA

Displays information about the current instance's System Global Area. Note, you need `SELECT ON V_$SGA` object privileges to use the `SGA` clause, otherwise you will receive a message

ORA-00942: table or view does not exist

SQLCODE

Shows the value of `SQL.SQLCODE` (the SQL return code of the most recent operation).

TTI[TLE]

Shows the current TTITLE definition.

USER

Shows the username you are currently using to access SQL*Plus. If you connect as `"/ AS SYSDBA"`, then the `SHOW USER` command displays

USER is "SYS"

Examples



To list the current `LINESIZE`, enter

```
SHOW LINESIZE
```



If the current linesize is 80 characters, SQL*Plus will give the following response:

```
LINESIZE 80
```

The following example illustrates how to create a stored procedure and then show its compilation errors:



```
CONNECT SYSTEM/MANAGER
CREATE PROCEDURE HR.PROC1 AS
BEGIN
:P1 := 1;
END;
/
```



```
Warning: Procedure created with compilation errors.
```



```
SHOW ERRORS
```



```
Errors for PROCEDURE HR.PROC1:
LINE/COL ERROR
```

```
-----
3/1      PLS-00049: bad bind variable 'P1'
```

```
SHOW ERRORS PROCEDURE PROC1
```

```
NO ERRORS.
```

```
SHOW ERRORS PROCEDURE HR.PROC1
```

```
Errors for PROCEDURE HR.PROC1:
```

```
LINE/COL ERROR
```

```
-----
3/3      PLS-00049: bad bind variable 'P1'
```

To show whether `AUTORECOVERY` is enabled, enter



```
SHOW AUTORECOVERY
```



```
AUTORECOVERY ON
```

To display the connect identifier for the default instance, enter



```
SHOW INSTANCE
```



```
INSTANCE "LOCAL"
```

To display the location for archive logs, enter



```
SHOW LOGSOURCE
```



```
LOGSOURCE "/usr/oracle90/dbs/arch"
```

To display information about the SGA, enter



```
SHOW SGA
```



```
Total System Global Area    7629732 bytes
Fixed Size                    60324 bytes
Variable Size                 6627328 bytes
Database Buffers             409600 bytes
Redo Buffers                  532480 bytes
```

SHUTDOWN

Syntax

SHUTDOWN [ABORT|IMMEDIATE|NORMAL|TRANSACTIONAL [LOCAL]]

Shuts down a currently running Oracle instance, optionally closing and dismounting a database. You cannot use SHUTDOWN to stop Oracle instances on Oracle7 servers.

Terms

Refer to the following list for a description of each term or clause:

ABORT

Proceeds with the fastest possible shutdown of the database without waiting for calls to complete or users to disconnect.

Uncommitted transactions are not rolled back. Client SQL statements currently being processed are terminated. All users currently connected to the database are implicitly disconnected and the next database startup will require instance recovery.

You must use this option if a background process terminates abnormally.

IMMEDIATE

Does not wait for current calls to complete or users to disconnect from the database.

Further connects are prohibited. The database is closed and dismounted. The instance is shutdown and no instance recovery is required on the next database startup.

NORMAL

NORMAL is the default option which waits for users to disconnect from the database.

Further connects are prohibited. The database is closed and dismounted. The instance is shutdown and no instance recovery is required on the next database startup.

TRANSACTIONAL [LOCAL]

Performs a planned shutdown of an instance while allowing active transactions to complete first. It prevents clients from losing work without requiring all users to log off.

No client can start a new transaction on this instance. Attempting to start a new transaction results in disconnection. After completion of all transactions, any client still connected to the instance is disconnected. Now the instance shuts down just as it would if a `SHUTDOWN IMMEDIATE` statement was submitted. The next startup of the database will not require any instance recovery procedures.

The `LOCAL` mode specifies a transactional shutdown on the local instance only, so that it only waits on local transactions to complete, not all transactions. This is useful, for example, for scheduled outage maintenance.

Usage

`SHUTDOWN` with no arguments is equivalent to `SHUTDOWN NORMAL`.

You must be connected to a database as `SYSOPER`, or `SYSDBA`. You cannot connect via a multi-threaded server. For more information about connecting to a database, see the `CONNECT` command earlier in this chapter.

Examples



To shutdown the database in normal mode, enter

```
SHUTDOWN
```



```
Database closed.  
Database dismounted.  
Oracle instance shut down.
```

START

Syntax

START {*uri|file_name[.ext]*} [*arg...*]

Runs the SQL*Plus statements in the specified command file. The command file can be called from the local file system or from a web server. *uri* is only supported on Windows platforms in this release. Only the *uri* form is supported in *iSQL*Plus*.

Terms

Refer to the following list for a description of each term or clause:

uri

Specifies the Uniform Resource Identifier of a script to run on the specified web server. SQL*Plus supports HTTP, FTP and gopher protocols.

file_name[.ext]

Represents the command file you wish to execute. The file can contain any command that you can run interactively.

If you do not specify an extension, SQL*Plus assumes the default command-file extension (normally SQL).

When you enter `START file_name.ext`, SQL*Plus searches for a file with the filename and extension you specify in the current default directory. If SQL*Plus does not find such a file, SQL*Plus will search a system-dependent path to find the file. Some operating systems may not support the path search. Consult the Oracle installation and user's manual(s) provided for your operating system for specific information related to your operating system environment.

arg ...

Represent data items you wish to pass to parameters in the command file. If you enter one or more arguments, SQL*Plus substitutes the values into the parameters (&1, &2, and so forth) in the command file. The first argument replaces each occurrence of &1, the second replaces each occurrence of &2, and so forth.

The START command DEFINES the parameters with the values of the arguments; if you START the command file again in this session, you can enter new arguments or omit the arguments to use the old values.

Usage

The @ (“at” sign) and @@ (double “at” sign) commands function similarly to START. Disabling the START command in the Product User Profile also disables the @ and @@ commands. See the @ (“at” sign) and @@ (double “at” sign) commands in this chapter for further information on these commands.

Examples

A file named PROMOTE with the extension SQL, used to promote employees, might contain the following command:



```
SELECT FIRST_NAME, LAST_NAME, JOB_ID, SALARY
FROM EMP_DETAILS_VIEW
WHERE JOB_ID='&1' AND SALARY>&2;
```

To run this command file, enter



```
START PROMOTE ST_MAN 7000
```

or if it is located on a web server, enter a command in the form:



```
START HTTP://HOST.DOMAIN/PROMOTE.SQL ST_MAN 7000
START FTP://HOST.DOMAIN/PROMOTE.SQL ST_MAN 7000
START GOPHER://HOST.DOMAIN/PROMOTE.SQL ST_MAN 7000
```

Where HOST.DOMAIN must be replaced by the host.domain name for the web server where the script is located.

In either case, SQL*Plus then executes the following command:

```
SELECT LAST_NAME, LAST_NAME
FROM EMP_DETAILS_VIEW
WHERE JOB_ID='ST_MAN' AND SALARY>7000;
```

and displays the results in SQL*Plus.

STARTUP

Syntax

```
STARTUP [FORCE] [RESTRICT] [PFILE=filename] [MOUNT [dbname]  
| OPEN [open_options] [dbname] | NOMOUNT]
```

where *open_options* has the following syntax:

```
READ {ONLY | WRITE [RECOVER]} | RECOVER
```

Starts an Oracle instance with several options, including mounting and opening a database. You cannot use STARTUP to start Oracle7 instances.

Terms

Refer to the following list for a description of each term and clause:

FORCE

Shuts down the current Oracle instance (if it is running) with SHUTDOWN mode ABORT, before restarting it. If the current instance is running and FORCE is not specified, an error results. FORCE is useful while debugging and under abnormal circumstances. It should not normally be used.

RESTRICT

Only allows Oracle users with the RESTRICTED SESSION system privilege to connect to the database. Later, you can use the ALTER SYSTEM command to disable the restricted session feature.

PFILE=*filename*

Causes the specified parameter file to be used while starting up the instance. If PFILE is not specified, then the default STARTUP parameter file is used. The default file used is platform specific. For example, the default file is \$ORACLE_HOME/dbs/init\$ORACLE_SID.ora on UNIX, and %ORACLE_HOME%\database\initORCL.ora on Windows.

MOUNT *dbname*

Mounts a database but does not open it.

dbname is the name of the database to mount or open. If no database name is specified, the database name is taken from the initialization parameter DB_NAME.

OPEN

Mounts and opens the specified database.

NOMOUNT

Causes the database not to be mounted upon instance startup.

Cannot be used with MOUNT, or OPEN.

RECOVER

Specifies that media recovery should be performed, if necessary, before starting the instance. STARTUP RECOVER has the same effect as issuing the RECOVER DATABASE command and starting an instance. Only complete recovery is possible with the RECOVER option.

Recovery proceeds, if necessary, as if AUTORECOVERY is set to ON, regardless of whether or not AUTORECOVERY is enabled. If a redo log file is not found in the expected location, recovery continues as if AUTORECOVERY is disabled, by prompting you with the suggested location and name of the subsequent log files that need to be applied.

Usage

You must be connected to a database as SYSOPER, or SYSDBA. You cannot be connected via a multi-threaded server.

STARTUP with no arguments is equivalent to STARTUP OPEN.

STARTUP OPEN RECOVER mounts and opens the database even when recovery fails.

Examples

To start an instance using the standard parameter file, mount the default database, and open the database, enter



```
STARTUP
```

or enter



```
STARTUP OPEN database
```

To start an instance using the standard parameter file, mount the default database, and open the database, enter



```
STARTUP FORCE RESTRICT NOMOUNT
```

To start an instance using the parameter file TESTPARM without mounting the database, enter



```
STARTUP PFILE=testparm NOMOUNT
```

To shutdown a particular database, immediately restart and open it, allow access only to database administrators, and use the parameter file MYINIT.ORA. enter



```
STARTUP FORCE RESTRICT PFILE=myinit.ora OPEN database
```

To startup an instance and mount but not open a database, enter



```
CONNECT / as SYSDBA
```



```
Connected to an idle instance.
```



```
STARTUP MOUNT
```



```
ORACLE instance started.
```

Total System Global Area	7629732 bytes
Fixed Size	60324 bytes
Variable Size	6627328 bytes
Database Buffers	409600 bytes
Redo Buffers	532480 bytes

TIMING

Syntax

TIMI[NG] [START *text*|SHOW|STOP]

Records timing data for an elapsed period of time, lists the current timer's name and timing data, or lists the number of active timers.

Terms

Refer to the following list for a description of each term or clause:

START *text*

Sets up a timer and makes *text* the name of the timer. You can have more than one active timer by STARTing additional timers before STOPping the first; SQL*Plus nests each new timer within the preceding one. The timer most recently STARTed becomes the current timer.

SHOW

Lists the current timer's name and timing data.

STOP

Lists the current timer's name and timing data, then deletes the timer. If any other timers are active, the next most recently STARTed timer becomes the current timer.

Enter TIMING with no clauses to list the number of active timers. For other information about TIMING, see SET AUTOTRACE

Usage

You can use this data to do a performance analysis on any commands or blocks run during the period.

For information about the data TIMING displays, see the Oracle installation and user's manual(s) provided for your operating system. Refer to the SET TIMING command for information on automatically displaying timing data after each SQL command or PL/SQL block you run.

To delete all timers, use the CLEAR TIMING command.

Examples



To create a timer named `SQL_TIMER`, enter

```
TIMING START SQL_TIMER
```



To list the current timer's title and accumulated time, enter

```
TIMING SHOW
```



To list the current timer's title and accumulated time and to remove the timer, enter

```
TIMING STOP
```

TTITLE

Syntax

TTI[TLE] [*printspec* [*text|variable*] ...] [ON|OFF]

where *printspec* represents one or more of the following clauses used to place and format the *text*:

COL *n*
S[KIP] [*n*]
TAB *n*
LE[FT]
CE[NTER]
R[IGHT]
BOLD
FORMAT *text*

Places and formats a specified title at the top of each report page or lists the current TTITLE definition. The old form of TTITLE is used if only a single word or string in quotes follows the TTITLE command.

Terms

Refer to the following list for a description of each term or clause. These terms and clauses also apply to the BTITLE command.

text

Represents the title text. Enter *text* in single quotes if you want to place more than one word on a single line.

variable

Represents a user variable or any of the following system-maintained values, SQL.LNO (the current line number), SQL.PNO (the current page number), SQL.RELEASE (the current Oracle release number), SQL.SQLCODE (the current error code), or SQL.USER (the current username).

To print one of these values, reference the appropriate variable in the title. You can format *variable* with the FORMAT clause.

OFF

Turns the title off (suppresses its display) without affecting its definition.

ON

Turns the title on (restores its display). When you define a top title, SQL*Plus automatically sets TTITLE to ON.

COL *n*

Indents to column *n* of the current line (backward if column *n* has been passed). “Column” in this context means print position, not table column.

S[KIP] [*n*]

Skips to the start of a new line *n* times; if you omit *n*, one time; if you enter zero for *n*, backward to the start of the current line.

TAB *n*

Skips forward *n* columns (backward if you enter a negative value for *n*). “Column” in this context means print position, not table column.

LE[FT]|CE[NTER]|R[IGHT]

Left-align, center, and right-align data on the current line respectively. SQL*Plus aligns following data items as a group, up to the end of the *printspec* or the next LEFT, CENTER, RIGHT, or COL command. CENTER and RIGHT use the SET LINESIZE value to calculate the position of the data item that follows.

BOLD

Prints data in bold print. SQL*Plus represents bold print on your terminal by repeating the data on three consecutive lines. On some operating systems, SQL*Plus may instruct your printer to print bold text on three consecutive lines, instead of bold.

FORMAT *text*

Specifies a format model that determines the format of following data items, up to the next FORMAT clause or the end of the command. The format model must be a *text* constant such as A10 or \$999. See the COLUMN FORMAT command for more information on formatting and valid format models.

If the datatype of the format model does not match the datatype of a given data item, the `FORMAT` clause has no effect on that item.

If no appropriate `FORMAT` model precedes a given data item, `SQL*Plus` prints `NUMBER` values according to the format specified by `SET NUMFORMAT` or, if you have not used `SET NUMFORMAT`, the default format. `SQL*Plus` prints `DATE` values according to the default format.

Refer to the `FORMAT` clause of the `COLUMN` command in this chapter for more information on default formats.

Enter `TTITLE` with no clauses to list the current `TTITLE` definition.

Usage

If you do not enter a *printspec* clause before the first occurrence of *text*, `TTITLE` left justifies the text. `SQL*Plus` interprets `TTITLE` in the new form if a valid *printspec* clause (`LEFT`, `SKIP`, `COL`, and so on) immediately follows the command name.

See `COLUMN NEW_VALUE` for information on printing column and `DATE` values in the top title.

You can use any number of constants and variables in a *printspec*. `SQL*Plus` displays the constants and variables in the order you specify them, positioning and formatting each constant or variable as specified by the *printspec* clauses that precede it.

The length of the title you specify with `TTITLE` cannot exceed 2400 characters.

The continuation character (a hyphen) will not be recognized inside a single-quoted title text string. To be recognized, the continuation character must appear outside the quotes, as follows:

```
TTITLE CENTER 'Summary Report for' -  
> 'the Month of May'
```

Examples

To define “Monthly Analysis” as the top title and to left-align it, to center the date, to right-align the page number with a three-digit format, and to display “Data in Thousands” in the center of the next line, enter



```
TTITLE LEFT 'Monthly Analysis' CENTER '01 Jan 2001' -  
RIGHT 'Page:' FORMAT 999 SQL.PNO SKIP CENTER -  
'Data in Thousands'
```



Monthly Analysis

01 Jan 2001

Page: 1

Data in Thousands

To suppress the top title display without changing its definition, enter



TTITLE OFF

UNDEFINE

Syntax

UNDEF[INE] *variable* ...

Deletes one or more user variables that you defined either explicitly (with the DEFINE command) or implicitly (with an argument to the START command).

Terms

Refer to the following for a description of the term or clause:

variable

Represents the name of the user variable you wish to delete. One or more user variables may be deleted in the same command.

Examples

To undefine a user variable named POS, enter



```
UNDEFINE POS
```

To undefine two user variables named MYVAR1 and MYVAR2, enter



```
UNDEFINE MYVAR1 MYVAR2
```

VARIABLE

Syntax

```
VAR[iable] [variable [NUMBER|CHAR|CHAR (n [CHAR|BYTE])|NCHAR|NCHAR (n)  
|VARCHAR2 (n [CHAR|BYTE])|NVARCHAR2 (n)|CLOB|NCLOB|REFCURSOR]]
```

Declares a bind variable that can then be referenced in PL/SQL. For more information about PL/SQL, see your *PL/SQL User's Guide and Reference*.

VARIABLE without arguments displays a list of all the variables declared in the session. VARIABLE followed only by a variable name lists that variable.

Terms

Refer to the following list for a description of each term or clause:

variable

Represents the name of the bind variable you wish to create.

NUMBER

Creates a variable of type NUMBER with fixed length.

CHAR

Creates a variable of type CHAR (character) with length one.

CHAR (*n*[CHAR|BYTE])

Creates a variable of type CHAR having a length of *n* bytes or *n* characters. The maximum that *n* can be is 2000 bytes, and the minimum is 1 byte or 1 character. The maximum *n* for a CHAR variable with character semantics is determined by the number of bytes required to store each character for the chosen character set, with an upper limit of 2000 bytes. The length semantics are determined by the length qualifiers CHAR or BYTE, and if not explicitly stated, the value of the NLS_LENGTH_SEMANTICS environment variable is applied to the bind variable. Explicitly stating the length semantics at variable definition stage will always take precedence over the NLS_LENGTH_SEMANTICS setting.

NCHAR

Creates a variable of type NCHAR (national character) with length one.

NCHAR (*n*)

Creates a variable of type NCHAR having a length of *n* characters. The maximum that *n* can be is determined by the number of bytes required to store each character for the chosen national character set, with an upper limit of 2000 bytes. The only exception to this is when a SQL*Plus session is connected to a pre Oracle9i server, or the SQLPLUSCOMPATIBILITY system variable is set to a version less than 9.0.0. In this case the length *n* can be in bytes or characters depending on the chosen national character set, with the upper limit of 2000 bytes still retained.

VARCHAR2 (*n*[CHAR|BYTE])

Creates a variable of type VARCHAR2 having a length of up to *n* bytes or *n* characters. The maximum that *n* can be is 4000 bytes, and the minimum is 1 byte or 1 character. The maximum *n* for a VARCHAR2 variable with character semantics is determined by the number of bytes required to store each character for the chosen character set, with an upper limit of 4000 bytes. The length semantics are determined by the length qualifiers CHAR or BYTE, and if not explicitly stated, the value of the NLS_LENGTH_SEMANTICS environment variable is applied to the bind variable. Explicitly stating the length semantics at variable definition stage will always take precedence over the NLS_LENGTH_SEMANTICS setting.

NVARCHAR2 (*n*)

Creates a variable of type NVARCHAR2 having a length of up to *n* characters. The maximum that *n* can be is determined by the number of bytes required to store each character for the chosen national character set, with an upper limit of 4000 bytes. The only exception to this is when a SQL*Plus session is connected to a pre Oracle9i server, or the SQLPLUSCOMPATIBILITY system variable is set to a version less than 9.0.0. In this case the length *n* can be in bytes or characters depending on the chosen national character set, with the upper limit of 4000 bytes still retained.

CLOB

Creates a variable of type CLOB.

NCLOB

Creates a variable of type NCLOB.

REFCURSOR

Creates a variable of type REF CURSOR.

Usage

Bind variables may be used as parameters to stored procedures, or may be directly referenced in anonymous PL/SQL blocks.

To display the value of a bind variable created with VARIABLE, use the PRINT command. For more information, see the PRINT command in this chapter.

To automatically display the value of a bind variable created with VARIABLE, use the SET AUTOPRINT command. For more information, see the SET AUTOPRINT command in this chapter.

Bind variables cannot be used in the COPY command or SQL statements, except in PL/SQL blocks. Instead, use substitution variables.

When you execute a VARIABLE ... CLOB or NCLOB command, SQL*Plus associates a LOB locator with the bind variable. The LOB locator is automatically populated when you execute a SELECT clob_column INTO :cv statement in a PL/SQL block. SQL*Plus closes the LOB locator after completing a PRINT statement for that bind variable, or when you exit SQL*Plus.

SQL*Plus SET commands such as SET LONG and SET LONGCHUNKSIZE and SET LOBOFFSET may be used to control the size of the buffer while PRINTing CLOB or NCLOB bind variables.

SQL*Plus REFCURSOR bind variables may be used to reference PL/SQL 2.3 or higher Cursor Variables, allowing PL/SQL output to be formatted by SQL*Plus. For more information on PL/SQL Cursor Variables, see your *PL/SQL User's Guide and Reference*.

When you execute a VARIABLE ... REFCURSOR command, SQL*Plus creates a cursor bind variable. The cursor is automatically opened by an OPEN ... FOR SELECT statement referencing the bind variable in a PL/SQL block. SQL*Plus closes the cursor after completing a PRINT statement for that bind variable, or on exit.

SQL*Plus formatting commands such as BREAK, COLUMN, COMPUTE and SET may be used to format the output from PRINTing a REFCURSOR.

A REFCURSOR bind variable may not be PRINTed more than once without re-executing the PL/SQL OPEN...FOR statement.

Examples

The following example illustrates creating a bind variable and then setting it to the value returned by a function:



```
VARIABLE id NUMBER
BEGIN
  :id := EMP_MANAGEMENT.HIRE
  ('BLAKE', 'MANAGER', 'KING', 2990, 'SALES');
END;
/
```

The value returned by the stored procedure is being placed in the bind variable, :id. It can be displayed with the PRINT command or used in subsequent PL/SQL subprograms.

The following example illustrates automatically displaying a bind variable:



```
SET AUTOPRINT ON
VARIABLE a REFCURSOR
BEGIN
  OPEN :a FOR SELECT LAST_NAME, CITY, DEPARTMENT_ID
  FROM EMP_DETAILS_VIEW
  WHERE SALARY > 12000
  ORDER BY DEPARTMENT_ID;
END;
/
```



PL/SQL procedure successfully completed.

LAST_NAME	CITY	DEPARTMENT_ID
Hartstein	Toronto	20
Russell	Oxford	80
Partners	Oxford	80
King	Seattle	90
Kochhar	Seattle	90
De Haan	Seattle	90

6 rows selected.

In the above example, there is no need to issue a PRINT command to display the variable.

The following example creates some variables:



```
VARIABLE id NUMBER
VARIABLE txt CHAR (20)
VARIABLE myvar REFCURSOR
```

Enter VARIABLE with no arguments to list the defined variables:



```
VARIABLE
```



```
variable id
datatype NUMBER
```

```
variable txt
datatype CHAR(20)
```

```
variable myvar
datatype REFCURSOR
```

The following example lists a single variable:



```
VARIABLE txt
```



```
variable txt
datatype CHAR(20)
```

The following example illustrates producing a report listing individual salaries and computing the departmental salary cost for employees who earn more than \$12,000 per month:



```
VARIABLE rc REFCURSOR
BEGIN
  OPEN :rc FOR SELECT DEPARTMENT_NAME, LAST_NAME, SALARY
  FROM EMP_DETAILS_VIEW
  WHERE SALARY > 12000
  ORDER BY DEPARTMENT_NAME, LAST_NAME;
END;
/
```



```
PL/SQL procedure successfully completed.
```



```
SET PAGESIZE 100 FEEDBACK OFF
TTITLE LEFT '*** Departmental Salary Bill ***' SKIP 2
COLUMN SALARY FORMAT $999,990.99 HEADING 'Salary'
COLUMN DEPARTMENT_NAME HEADING 'Department'
COLUMN LAST_NAME HEADING 'Employee'
```

```

COMPUTE SUM LABEL 'Subtotal:' OF SALARY ON DEPARTMENT_NAME
COMPUTE SUM LABEL 'Total:' OF SALARY ON REPORT
BREAK ON DEPARTMENT_NAME SKIP 1 ON REPORT SKIP 1
PRINT rc

```



```

*** Departmental Salary Bill ***

```

DEPARTMENT_NAME	Employee	Salary

Executive	De Haan	\$17,000.00
	King	\$24,000.00
	Kochhar	\$17,000.00

Subtotal:		\$58,000.00

Marketing	Hartstein	\$13,000.00

Subtotal:		\$13,000.00

Sales	Partners	\$13,500.00
	Russell	\$14,000.00

Subtotal:		\$27,500.00

Total:		\$98,500.00

The following example illustrates producing a report containing a CLOB column, and then displaying it with the SET LOBOFFSET command.

Assume you have already created a table named `clob_tab` which contains a column named `clob_col` of type CLOB. The `clob_col` contains the following data:

Remember to run the Departmental Salary Bill report each month. This report contains confidential information.

To produce a report listing the data in the `col_clob` column, enter



```

VARIABLE T CLOB
BEGIN
  SELECT CLOB_COL INTO :T FROM CLOB_TAB;
END;
/
PL/SQL PROCEDURE SUCCESSFULLY COMPLETED

```



To print 200 characters from the column `clob_col`, enter



```
SET LINESIZE 70
SET LONG 200
PRINT T
```



```
T
```

Remember to run the Departmental Salary Bill report each month. This report contains confidential information.

To set the printing position to the 21st character, enter



```
SET LOBOFFSET 21
PRINT T
```



```
T
```

Departmental Salary Bill report each month. This report contains confidential information.

For more information on creating CLOB columns, see your *Oracle9i SQL Reference*.

WHENEVER OSERROR

Syntax

WHENEVER OSERROR {CONTINUE [COMMIT|ROLLBACK|NONE]}

Performs the specified action if an operating system error occurs (such as a file I/O error).

Terms

Refer to the following list for a description of each term or clause:

CONTINUE

Turns off the EXIT option. EXIT is not supported in this release of *iSQL*Plus*.

COMMIT

Directs *SQL*Plus* to execute a COMMIT before exiting or continuing and save pending changes to the database.

ROLLBACK

Directs *SQL*Plus* to execute a ROLLBACK before exiting or continuing and abandon pending changes to the database.

NONE

Directs *SQL*Plus* to take no action before continuing.

Usage

If you do not enter the **WHENEVER OSERROR** command, the default behavior of *SQL*Plus* is to continue and take no action when an operating system error occurs.

If you do not enter the **WHENEVER SQLERROR** command, the default behavior of *SQL*Plus* is to continue and take no action when a SQL error occurs.

WHENEVER SQLERROR

Syntax

WHENEVER SQLERROR {CONTINUE [COMMIT|ROLLBACK|NONE]}

Performs the specified action if a SQL command or PL/SQL block generates an error.

Terms

Refer to the following list for a description of each term or clause:

CONTINUE

Turns off the EXIT option. EXIT is not supported in this release of iSQL*Plus.

COMMIT

Directs SQL*Plus to execute a COMMIT before exiting or continuing and save pending changes to the database.

ROLLBACK

Directs SQL*Plus to execute a ROLLBACK before exiting or continuing and abandon pending changes to the database.

NONE

Directs SQL*Plus to take no action before continuing.

Usage

The WHENEVER SQLERROR command is triggered by SQL command or PL/SQL block errors, and not by SQL*Plus command errors.

SQL*Plus Error Messages

This appendix lists error messages generated by SQL*Plus and iSQL*Plus. For error messages generated by Oracle, refer to the *Oracle9i Error Messages*.

This chapter contains information about:

- SQL*Plus Error Messages and Codes
- iSQL*Plus Error Messages
- COPY Command Messages

SQL*Plus Error Messages and Codes

SP2-0002 ACCEPT statement must specify a variable name.

Cause: Required variable name was missing after the ACCEPT command.

Action: Re-enter the ACCEPT command with a variable argument to store the input value.

SP2-0003 Ill-formed ACCEPT command starting as *command_string*

Cause: An invalid option was used in the ACCEPT command.

Action: Check the syntax of the ACCEPT command in Chapter 5, "Command Reference" for the correct option.

SP2-0004 Nothing to append

Cause: There was no specified text entered after the APPEND command.

Action: Re-enter the APPEND command with the specified text.

SP2-0006 Not enough room to format computations

Cause: Unable to allocate memory to format computations.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0015 No break(s) defined

Cause: There was no break defined.

Action: Define a break. Check the syntax of the BREAK command in Chapter 5, "Command Reference" for the correct options.

SP2-0016 Break specification must start with ON/BY or ACROSS keyword

Cause: An invalid option was used in the BREAK command.

Action: Check the syntax of the BREAK command in Chapter 5, "Command Reference" for the correct options.

SP2-0017 Missing column name after '*keyword_name*' keyword

Cause: There was no column name after the specified keyword.

Action: Enter a column name after the specified keyword.

SP2-0019 Invalid numeric argument to *option_name* option

Cause: An invalid numeric argument was used in the specified option.

Action: Correct the argument and try again.

SP2-0020 No storage available for *column_name*

Cause: An error has occurred. SQL*Plus was unable to allocate memory for a BREAK command.

Action: Allocate more memory by closing some applications.

SP2-0022 Cannot allocate space to modify the *buffer_name* buffer variable

Cause: An internal error occurred.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0023 String not found

Cause: The search string specified was not found.

Action: Check the search string to make sure that it is valid.

SP2-0024 Nothing to change

Cause: There was nothing in the SQL buffer when using the CHANGE command.

Action: Make sure the SQL buffer is not empty before using the CHANGE command.

SP2-0025 Invalid change string

Cause: An invalid option was used in the CHANGE command.

Action: Check the syntax of the CHANGE command in Chapter 5, "Command Reference" for the correct options.

SP2-0026 No lines to delete

Cause: There was nothing in the SQL buffer when using the DEL command.

Action: Make sure the SQL buffer is not empty before using the DEL command.

SP2-0027 Input is too long (> *max_characters* characters) - line ignored

Cause: The input value specified was too long.

Action: Re-enter with fewer characters.

SP2-0028 INTERNAL SQL*Plus ERROR - Invalid mode (*mode_number*)

Cause: An internal error occurred.

Action: Note the message and number, and contact Oracle Support Services.

SP2-0029 Command buffer space exhausted

Cause: A large SQL or PL/SQL script is being executed from SQL*Plus.

Action: Reduce the size of the SQL statement or PL/SQL block by one of the following:

- Remove extra white space and comments.
- Re-code to use fewer commands and/or shorter variable names.
- Place sections of the block into stored (or packaged) procedures, and then call these procedures from the block.

SP2-0030 No room for another line

Cause: The maximum number of lines in a SQL statement or PL/SQL block has been exceeded.

Action: Reduce the number of lines and try again.

SP2-0038 Command too long. (*max_characters* characters)

Cause: The specified command entered was too long.

Action: Check Chapter 5, "Command Reference" for the limitation.

SP2-0039 Command line overflow while substituting into *command_name*

Cause: The maximum length of the command line has been exceeded.

Action: Reduce the length of the substitution string.

SP2-0042 Unknown command *command_name* - rest of line ignored

Cause: The command entered was not valid.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct options of the command you used.

SP2-0044 For a list of known commands enter HELP and to leave enter EXIT

Cause: An unknown command was entered.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct options of the command you used.

SP2-0045 No *column_name* defined

Cause: No columns have been defined.

Action: No action required.

SP2-0046 *column_name* not defined

Cause: The column name specified was not defined.

Action: Retry again with a valid column name.

SP2-0047 Invalid number for *option_name* option

Cause: An invalid number was used for this option.

Action: Re-try the operation with a valid number.

SP2-0051 Switch value is *switch_value* and is not handled properly

Cause: An internal error occurred.

Action: Note the message and number, and contact Oracle Support Services.

SP2-0052 Like *column_name*, *column_name* not defined

Cause: The column which the format is based on was not defined.

Action: Use the COLUMN command to make sure the column the format is based on is defined first.

SP2-0054 No room to allocate *definition_name* definition. Ignored

Cause: Unable to allocate memory to process the COLUMN command.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

**SP2-0055 Out of room while allocating portion of new *definition_name*.
Old definition (if any) retained**

Cause: Unable to allocate memory to store the new definition.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0080 No COMPUTES currently defined

Cause: No COMPUTE definition.

Action: Define a COMPUTE. Check the syntax of the COMPUTE command in Chapter 5, "Command Reference" for the correct options.

SP2-0081 Maximum of *number* COMPUTE functions allowed at a time

Cause: The maximum of COMPUTE functions has been exceeded.

Action: Reduce the number of COMPUTE functions.

SP2-0082 No COMPUTE functions requested

Cause: No COMPUTE functions requested.

Action: No action required.

SP2-0083 Warning: COMPUTE option *function_name* specified *number* times

Cause: A label or a function was specified more than once.

Action: Remove the unnecessary labels or functions.

SP2-0084 COMPUTE ON keyword specified already

Cause: The ON keyword was specified more than once.

Action: Specify the ON keyword once in the command.

SP2-0085 COMPUTE OF keyword specified already

Cause: The OF keyword was specified more than once.

Action: Specify the OF keyword once in the command.

SP2-0087 No room to allocate COMPUTE control block for *column_name*

Cause: Unable to allocate memory to process the COMPUTE command.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0088 Missing *keyword_name* keyword.

Usage: STORE [SET *filename.ext*] [CRE[ATE] | REP[LACE] | APP[END]]

Cause: Missing a keyword in the statement.

Action: Check the syntax in Chapter 5, "Command Reference" for the options of the command you used, and use the keyword in the appropriate place.

SP2-0092 Missing columns for *keyword_name* keyword

Cause: The column name was not been specified for the keyword.

Action: Specify the column name and try again.

SP2-0096 No more room to allocate INTO variable *variable_name*

Cause: Unable to allocate memory to process the COMPUTE command.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0097 No storage to allocate ON column *column_name*

Cause: Unable to allocate memory to process the COMPUTE command.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0098 No storage to allocate COMPUTE block for *column_name*

Cause: Unable to allocate memory to process the COMPUTE command.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0103 Nothing in SQL buffer to run

Cause: Nothing was in the SQL buffer to run.

Action: Enter a valid SQL command.

SP2-0105 Illegal, or missing, entity name

Cause: File name was not specified in the GET or SAVE commands.

Action: Specify a file name and try again.

SP2-0107 Nothing to save

Cause: Nothing in the SQL buffer when attempting to save the content to a file.

Action: Enter a SQL command to save.

SP2-0108 The names CREATE, REPLACE, APPEND, and abbreviations may not be used

Cause: The file name specified was the word "file".

Action: Put the name in single quotes.

SP2-0109 Cannot append to file *file_name*

Cause: An attempt was made to append the content of the SQL buffer to a file and the file could not be written. Possible causes:

- An error was encountered when creating the destination file.
- A directory name specified in the SAVE statement was not found.
- A system error made it impossible to open the file.

Action: Take the following actions

- Check that the destination is valid and that there is sufficient space on the destination device.
- Check the statement for a typing mistake in the directory name. Then issue the statement again after correcting the directory name.

SP2-0110 Cannot create save file *file_name*

Cause: An attempt was made to save the content of the SQL buffer to a file and the file could not be written. Possible causes:

- An error was encountered when creating the destination file.
- A directory name specified in the SAVE statement was not found.
- A system error made it impossible to open the file.

Action: Take the following actions:

- Check that the destination is valid and that there is sufficient space on the destination device.
- Check the statement for a typing mistake in the directory name. Then issue the statement again after correcting the directory name.

SP2-0111 Cannot close save file *file_name*

Cause: The file was in use.

Action: Release the file from the other process.

SP2-0116 Illegal SAVE command

Cause: An invalid option was used in the SAVE command.

Action: Check the syntax of the SAVE command in Chapter 5, "Command Reference" for the correct options.

SP2-0134 No symbols currently defined

Cause: No DEFINE symbols were defined.

Action: No action required.

SP2-0135 Symbol *symbol_name* is UNDEFINED

Cause: The specified symbol was undefined.

Action: Re-enter the DEFINE command with an assignment clause or a valid symbol or variable name.

SP2-0136 DEFINE requires an equal sign (=)

Cause: Expecting an equal sign after a symbol or variable name in the DEFINE command.

Action: Specify an equal sign after the symbol or variable name.

SP2-0137 DEFINE requires a value following equal sign

Cause: There was no value for the variable or symbol. SQL*Plus expected a value to be assigned to a symbol or variable name after the equal sign.

Action: Specify a value for the symbol or variable.

SP2-0138 DEFINE *variable* not added (no room)

Cause: Maximum number of variables that can be defined in a SQL*Plus session was exceeded.

Action: UNDEFINE any unused variables to make room for this variable and re-run the command.

SP2-0145 Udalnk is not 12345. Probably a link error

Action: The SQL*Plus executable is not linked correctly.

Action: Make a note of the message and the number, then contact the System Administrator to re-link SQL*Plus.

SP2-0146 Unable to allocate dynamic space needed (*number_of_bytes* bytes) - exiting

Cause: An internal error occurred.

Action: Note the message and number, and contact the System Administrator.

SP2-0152 ORACLE may not be functioning properly

Cause: Unable to initialize a session to the Oracle instance.

Action: Make a note of the message and the number, then contact Database Administrator.

SP2-0157 Unable to CONNECT to ORACLE after 3 attempts, exiting SQL*Plus

Cause: Unable to connect to Oracle after three attempts.

Action: Validate login details and re-try.

SP2-0158 Unknown *command_name* option "*option_name*"

Usage: SET SHIFT[INOUT] [VIS[IBLE | INV[ISIBLE]]
SET NEWP[AGE] [1 | n | NONE]

Cause: An invalid option was specified for the given command.

Action: Check the syntax of the command in Chapter 5, "Command Reference" for the correct options.

SP2-0160 Unable to open *file_name*

Cause: Possible causes:

- The file was not found under the specified name in the specified location.
- File lacked the necessary privileges to open the file.
- A system error made it impossible to open the file.

Action: Take the following actions:

- Make sure the file name specified is stored in the appropriate directory.
- Make sure that the file has the privileges necessary for access. If it does not then change privileges accordingly.
- Consult operating system documentation or contact the System Administrator.

SP2-0161 Line *line_number* truncated

Cause: The line in the file was too long.

Action: No action required or reduce the length of the line.

SP2-0162 Unable to close *file_name*

Cause: Unable to close the specified file as it was being used.

Action: Release the file from the other process.

SP2-0171 HELP not accessible

Cause: On-line SQL*Plus help is not installed in this Oracle instance.

Action: Contact the Database Administrator to install the on-line help.

SP2-0172 No HELP available

Cause: There is no help information available for the specified command.

Action: Contact the Database Administrator to install the help system.

SP2-0176 Option ? Is invalid

Cause: The option ? is not a valid in this command.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct options for the command you used.

SP2-0187 Error in variable assignment

Cause: The assignment for the specified variable was incorrect.

Action: Refer to the ACCEPT command in Chapter 5, "Command Reference" and correct the syntax.

SP2-0223 No lines in *buffer_name* buffer

Cause: There are no lines stored in the buffer.

Action: Enter SQL statements into the buffer.

SP2-0224 Invalid starting line number

Cause: The line number specified was incorrect.

Action: Check and make sure that the line number is correct and try again.

SP2-0225 Invalid ending line number

Cause: The line number specified was incorrect.

Action: Check and make sure that the line number is correct and try again.

SP2-0226 Invalid line number *current_line_number*

Cause: Invalid line number was specified.

Action: Re-enter with a valid line number.

SP2-0232 Input too long. Must be less than *number_of_characters* characters

Cause: The input value was too long.

Action: Reduce the size of the value and re-enter.

SP2-0233 Unable to obtain userid after *number_of_attempts* attempts. Retry command

Cause: SQL*Plus was unable to login after three attempts.

Action: Make sure the userid and password is correct and try again.

SP2-0240 Enter value for *variable_name*:

Cause: A substitution variable was used and SQL*Plus was unable to find a value for that variable.

Action: Enter a value at the prompt for the substitution variable.

SP2-0241 No room for symbol *symbol_name*:(not defined)

Cause: Unable to allocate memory for the symbol.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0244 Cannot issue a PRINT command within a PAGE break

Cause: The PRINT command is not allowed within a PAGE break.

Action: Check Chapter 5, "Command Reference" for the correct syntax.

SP2-0245 Unable to allocate temporary storage for printing

Cause: Unable to allocate temporary storage for printing.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0246 Illegal FORMAT string *column_format_name*

Cause: An invalid format was specified for the column.

Action: Specify a valid format for the column.

SP2-0249 *variable_name* not a valid variable type for printing

Cause: The specified variable is not valid for printing.

Action: Check the variable type before re-typing the command.

SP2-0253 Data item *line_number (data_item_name)* will not fit on line

Cause: The current line size setting is too small to fit the specified data item on a line.

Action: Increase the line size so that the item can be displayed.

SP2-0258 Could not create variable *variable_name* for column *column_name*

Cause: The specified variable could not be created for column – internal error or out of memory.

Action: Check memory usage.

SP2-0259 Could not create variable *variable_name* for COMPUTE INTO

Cause: The specified variable could not be created.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct option of the command you used.

SP2-0260 Computation for column *column_name* not uniquely qualified.

Could be for table *table_name* or *table_name*. Computation ignored

Cause: The specified column was not uniquely qualified in the statement.

Action: Check syntax in Chapter 5, "Command Reference" for the correct option of the command you used.

SP2-0262 No room to allocate CCBDEF pointer array

Cause: An internal memory error occurred.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

**SP2-0263 No room to allocate COMPUTE block for *column_name*
ON *page/report/column_name***

Cause: Insufficient memory allocated to the COMPUTE block.

Action: Allocate more memory by closing other applications.

SP2-0265 *option_name* must be set ON or OFF

Cause: An invalid SET option name was specified.

Action: Re-enter with either ON or OFF as one of the SET options.

SP2-0266 Internal error: buffer (*buffer_size*) smaller than 1 (*buffer_limit*)

Cause: An internal error occurred.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0267 *option_name* option parameter_number (*lower_range* through *upper_range*)

Cause: A value for a parameter was out of the specified range.

Action: Check the limits of the parameter and enter a value that is within the limit.

SP2-0268 *option_name* option not a valid number

Cause: Non-numeric value (integer) was entered for a parameter.

Action: Enter a valid numeric value (integer).

SP2-0271 *variable_name* is not a buffer variable

Cause: The specified variable was not defined as a buffer.

Action: Make sure that the buffer variable name is correct and try again.

SP2-0272 *character_name* character cannot be alphanumeric or white-space

Cause: The specified character in the SET command cannot be alphanumeric or white-space.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct option of the command you used.

SP2-0277 *entered_value* value not valid

Cause: The value entered was incorrect.

Action: Re-enter with a valid value.

SP2-0281 *option_name* missing set option

Usage: SET SHIFT[INOUT] [VIS[IBLE | INV[ISIBLE]]
SET MARKUP HTML [ON | OFF] [HEAD *text*] [BODY *text*] [TABLE
text] [ENTMAP [ON | OFF]] [SPOOL [ON | OFF]] [PRE[FORMAT]
[ON | OFF]]
[-M[ARKUP] \ "HTML [ON | OFF] [HEAD *text*] [BODY *text*]

Cause: SET option was missing in the command.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct options of the command you used.

SP2-0306 Invalid option

Cause: Invalid option was specified for the command.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct options of the command you used.

SP2-0308 Cannot close spool file

Cause: The file is currently being used.

Action: Release the file from the other process.

SP2-0309 SQL*Plus command procedures may only be nested to a depth of *number_of_nested_procedures*

Cause: Maximum number of nested procedures or scripts was reached.

Action: Reduce the number of nested procedures or scripts.

SP2-0310 Unable to open file *file_name*

Cause: Unable to open the specified file.

Action: Check and make sure the file name is valid.

SP2-0311 String expected but not found

Cause: SQL*Plus was expecting a string at the end of the command, but could not find it.

Action: Retry the command with a valid string. Check the syntax in the Chapter 5, "Command Reference" for the correct options for the command you used.

SP2-0312 Missing terminating quote (*quote_type*)

Cause: The DESCRIBE command schema or object did not have a terminating quote.

Action: Close the opening quotation mark with the corresponding closing quotation mark.

SP2-0317 Expected symbol name is missing

Cause: SQL*Plus was expecting a symbol, but it was not specified.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct options for the command you used.

SP2-0318 Symbol name beginning *variable_name..* Is too long (*max_max_name_length*)

Cause: Specified variable name exceeded the maximum name length.

Action: Reduce the size of the symbol name and re-enter.

SP2-0323 No room to add timing element - request denied

Cause: Unable to allocate memory while trying to run the TIMING command.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0324 Operating system timing error *error_option_number* - request denied

Cause: The TIMING command failed to initialize due to a possible operating system error.

Action: Resolve the operating system error and try again.

SP2-0325 No timing elements to *option_name*

Cause: There are no timers recorded to SHOW or STOP.

Action: Check that timers were created with the TIMING command.

SP2-0328 No room to allocate title buffer

Cause: Unable to allocate memory while trying to run the TTITLE or BTITLE command.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0332 Cannot create spool file

Cause: Possible causes:

- Insufficient privileges to create a file.
- A system error made it impossible to create a file.

Action: Take the following actions:

- Change privileges to allow creation of the file.
- Consult the operating system documentation or contact the System Administrator.

SP2-0333 Illegal spool file name: "*spool_name*" (bad character: '*character_name*')

Cause: An invalid filename was entered in the SPOOL command.

Action: Correct the filename and re-enter.

SP2-0341 Line overflow during variable substitution (>*number_of_characters* characters at line *line_number*)

Cause: The maximum number of characters was exceeded in the SQL buffer after the substitution variable has been expanded.

Action: Reduce the length in the substitution variable and try again.

SP2-0357 Out of temporary storage

Cause: Unable to allocate memory while trying to run the command.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0359 Memory exhausted

Cause: Unable to allocate memory while trying to run the command.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0381 *command_name* is not available

Cause: The command specified is not implemented.

Action: Use the appropriate SQL*Plus command. Refer to Chapter 5, "Command Reference" for a list of commands and their correct syntax.

SP2-0382 The *command_name* command is not available

Cause: The command was not recognized, or it is disabled. This occurs if it is a command that does not have any meaning in SQL*Plus (such as a SQL buffer editing command), or it is not allowed for security reasons.

Action: Remove the command from the script. Refer to Chapter 5, "Command Reference" for a list of commands and their correct syntax.

SP2-0392 Cannot UNDEFINE the current edit buffer

Cause: The current edit buffer cannot be undefined.

Action: No action required.

SP2-0394 Illegal buffer name: *buffer_name*

Cause: An buffer name contained an illegal character, for example hyphen (-).

Action: Correct and remove the illegal character from the buffer name.

SP2-0395 Usage: SQLPLUS [[<option>] [<login>] [<start>]]

Where <option> ::= -H | -V | [[-M <o>] [-R <n>] [-S]]

-H displays the SQL*Plus version banner and usage syntax

-M <o> uses HTML markup options <o>

-V displays the SQL*Plus version banner

-S uses silent mode

-R <n> uses restricted mode <n>

<login> ::= <username>[/<password>][@<connect_string>] | / | /NOLOG

<start> ::= @<filename>[.<ext>] [<parameter> ...]

Cause: A SQL*Plus command option was invalid.

Action: Check the syntax for the SQLPLUS command in "Starting SQL*Plus and Getting Help" in Chapter 7 of the *SQL*Plus User's Guide and Reference* for the correct usage.

SP2-0423 Illegal GET command

Cause: An invalid option was used in the GET command.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct options of the command you used.

SP2-0425 value is not a valid number

Cause: The value entered in the ACCEPT command was not a number.

Action: Correct the entry and enter a valid number.

SP2-0426 Input truncated to *number_of_characters* characters

Cause: There was no carriage return at the last line of the SQL statement.

Action: Insert a carriage return.

SP2-0450 Usage: WHENEVER SQLERROR.

[CONTINUE [COMMIT | ROLLBACK | NONE] | EXIT [SUCCESS | FAILURE | WARNING | n | <variable> | :<bindvariable>] [COMMIT | ROLLBACK]]

Cause: An option to WHENEVER SQLERROR was invalid in SQL*Plus.

Action: Specify a valid option.

SP2-0453 Usage: WHENEVER OSERROR

{ CONTINUE [COMMIT | ROLLBACK | NONE] | EXIT [SUCCESS | FAILURE | WARNING | n | <variable> | :<bindvariable> | OSCODE] [COMMIT | ROLLBACK] }

Cause: An option to WHENEVER OSERROR was invalid in SQL*Plus.

Action: Specify a valid option.

SP2-0480 A missing FROM or TO clause uses the current SQL*Plus connection.

Usage: COPY FROM <db> TO <db> <opt> <table> [(<cols>)] USING <sel>

<db> : database string, e.g., hr/hr@d:chicago-mktg

<opt> : ONE of the keywords: APPEND, CREATE, INSERT or REPLACE.

<table>: name of the destination table.

<cols> : a comma-separated list of destination column aliases.

<sel> : any valid SQL SELECT statement.

Cause: Usage for COPY command was specified incorrectly.

Action: Specify a valid option.

SP2-0495 FROM and TO clauses both missing; specify at least one

Cause: The FROM and TO clauses were missing from the COPY statement.

Action: Specify at least one clause. Check the syntax in Chapter 5, "Command Reference" for the correct options of the command you used.

SP2-0496 Misplaced FROM clause

Cause: The FROM keyword was in the wrong position in the COPY command.

Action: Check the syntax of the COPY command in Chapter 5, "Command Reference" for the correct options.

SP2-0497 Misplaced TO clause

Cause: The TO keyword was in the wrong position in the COPY command.

Action: Check the syntax of the COPY command in Chapter 5, "Command Reference" for the correct options.

SP2-0498 Missing parenthetical column list or USING keyword

Cause: A parenthetical list was missing in the column list or the USING keyword is missing in the COPY command.

Action: Check the syntax of the COPY command in Chapter 5, "Command Reference" for the correct options.

SP2-0499 Misplaced APPEND keyword

Cause: The APPEND keyword was in the wrong position in the COPY command.

Action: Check the syntax of the COPY command in Chapter 5, "Command Reference" for the correct options.

SP2-0501 Error in SELECT statement: *Oracle_database_error_message*

Cause: Invalid SELECT statement found in the COPY command.

Action: Check the syntax of the COPY command in Chapter 5, "Command Reference" for the correct options.

SP2-0513 Misplaced CREATE keyword

Cause: The CREATE keyword was in the wrong position in the COPY command.

Action: Check the syntax of the COPY command in Chapter 5, "Command Reference" for the correct options.

SP2-0514 Misplaced REPLACE keyword

Cause: The REPLACE keyword was in the wrong position in the COPY command.

Action: Check the syntax of the COPY command in Chapter 5, "Command Reference" for the correct options.

SP2-0515 Maximum number of columns (*max_num_columns*) exceeded

Cause: An error occurred with the COPY command, maximum number of columns was exceeded in the command.

Action: Reduce the number of columns and try again.

SP2-0516 Invalid *command_name* name NULL encountered

Cause: Either COLUMN or ATTRIBUTE command used a null column name. Invalid column name specified in the command.

Action: Retry the operation with a valid column name.

SP2-0517 Missing comma or right parenthesis

Cause: A missing command right parenthesis was identified in the COPY command.

Action: Retry the operation with a comma or right parenthesis.

SP2-0518 Missing USING clause

Cause: USING keyword is missing in the USING clause of the COPY command.

Action: Specify the USING keyword before the query statement.

SP2-0519 FROM string missing Oracle Net @database specification

Cause: Missing connect string for the database that contains the data to be copied from in the COPY command.

Action: Re-specify the database. By omitting the FROM clause, the source defaults to the database to which SQL*Plus is connected. Include a FROM clause to specify a source database other than the default.

SP2-0520 TO string missing Net8 Oracle Net @database specification

Cause: Missing connect string for the database containing the destination table in the COPY command.

Action: Re-specify the database. By omitting the TO clause, the source defaults to the database to which SQL*Plus is connected. Include a TO clause to specify a source database other than the default.

SP2-0526 Misplaced INSERT keyword

Cause: The INSERT keyword is misplaced in the COPY command.

Action: Check the syntax of the COPY command in Chapter 5, "Command Reference" for the correct options.

SP2-0540 File *file_name* already exists. Use SAVE *filename*.[*ext*] REPLACE

Cause: The file specified already exists.

Action: Use the REPLACE option to overwrite the existing file, otherwise, specify another file name.

SP2-0545 SET command requires an argument

Cause: An argument was missing in the SET command.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct options of the command you used.

SP2-0546 User requested Interrupt or EOF detected

Cause: Either end-of-file was reached, or CTRL-C was entered to cancel the process.

Action: No action required.

SP2-0547 *option_name* option value out of range (*lower_value* through *upper_value*)

Cause: An out of range was reached for the specified SET option.

Action: Check the limits for the option and re-try the operation.

SP2-0548 Usage: VAR[IABLE] [<*variable*> [NUMBER | CHAR | CHAR (n [CHAR | BYTE]) | VARCHAR2 (n [CHAR | BYTE]) | NCHAR | NCHAR (n) | NVARCHAR2 (n) | CLOB | NCLOB | REFCURSOR]]

Cause: Usage message for VARIABLE command.

Action: Check the syntax of the VARIABLE command in Chapter 5, "Command Reference" for the correct usage.

SP2-0549 Usage: PRINT [<*variable*> ...]

Cause: Usage message for PRINT command.

Action: Check the syntax of the PRINT command in Chapter 5, "Command Reference" for the correct usage.

SP2-0550 Usage: SHOW ERRORS [[FUNCTION | PROCEDURE | PACKAGE | PACKAGE BODY | TRIGGER | VIEW | TYPE | TYPE BODY | JAVA SOURCE | JAVA CLASS] [*schema.*]*name*]

Cause: Usage message for SHOW ERRORS command.

Action: Check the syntax of the SHOW ERRORS command in Chapter 5, "Command Reference" for the correct options.

SP2-0552 Bind variable *variable_name* not declared

Cause: The specified bind variable was not declared.

Action: Check the search string to make sure that it is valid.

SP2-0556 Invalid file name

Cause: Missing file name or an invalid file name specified.

Action: Make sure that a file name was specified.

SP2-0559 EXEC[UTE] statement

Cause: Usage message of the EXECUTE command.

Action: Check the syntax of the EXECUTE command in Chapter 5, "Command Reference" for the correct usage.

SP2-0560 Usage: DESCRIBE [*schema.*]*object* [*subobject* | @*db_link*] [*column*]

Cause: Usage message of the DESCRIBE command.

Action: Check the syntax of the DESCRIBE command in Chapter 5, "Command Reference" for the correct usage.

SP2-0561 Object does not exist

Cause: The specified object you tried to DESCRIBE does not exist in the database.

Action: Retry the command with a valid object name.

SP2-0562 Object does not exist in package

Cause: The specified object you tried to DESCRIBE does not exist in the package.

Action: Check and make sure that the object name is correct.

SP2-0564 Object *object_name* is INVALID, it may not be described

Cause: The specified object you tried to DESCRIBE is invalid.

Action: Re-validate the object.

SP2-0565 Illegal identifier

Cause: Invalid character used in the DESCRIBE command.

Action: Correct the character and try again.

SP2-0566 Illegal sub-object specification

Cause: Invalid sub-object specification in the DESCRIBE command.

Action: Correct the subject specification and try again.

SP2-0567 Illegal column specification for PL/SQL object

Cause: It is illegal to describe a column within an object in the DESCRIBE command.

Action: Remove the column specification in the DESCRIBE command and try again.

SP2-0568 No bind variables declared

Cause: There are no bind variables declared.

Action: No action required.

SP2-0570 Usage: SET SERVEROUTPUT [ON | OFF] [SIZE [SIZE n] [FOR[MAT] [WRA[PPED] | WOR[D_WRAPPED] | TRU[NCATED]]]

Cause: An invalid option was used in the SET SERVEROUTPUT command.

Action: Specify a valid option.

SP2-0577 Usage: SET FLAGGER [OFF | ENTRY | INTERMEDIATE | FULL]

Cause: An invalid option was specified in the SET FLAGGER command.

Action: Specify a valid option.

SP2-0581 Object *object_name* is a package; use 'DESCRIBE *package.procedure*'

Cause: A package cannot be described as a stand-alone, it must be supplied with a sub-object, such as procedure.

Action: Use the DESCRIBE command to describe a sub-object within a package.

SP2-0582 Usage: [EXIT | QUIT] [SUCCESS | FAILURE | WARNING | n | <variable> | :<bindvariable>] [COMMIT | ROLLBACK]

Cause: An option to EXIT was invalid in SQL*Plus.

Action: Specify a valid option.

SP2-0584 EXIT variable *variable_name* was non-numeric

Cause: The specified EXIT variable is non-numeric.

Action: Check the syntax of the EXIT command in Chapter 5, "Command Reference" for the correct usage.

SP2-0590 A COMPUTE function must appear before each LABEL keyword

Cause: The function COMPUTE must appear before each LABEL keyword.

Action: Check the syntax of the COMPUTE command in Chapter 5, "Command Reference" for the correct usage.

**SP2-0591 Unable to allocate dynamic space needed (*number_of_bytes* bytes)
Try reducing ARRAYSIZE or the number of columns selected**

Cause: Unable to allocate memory to process the command.

Action: Free up additional memory by: closing applications not required; reducing the size of the command, or statement; or by recoding the query to select fewer records.

SP2-0593 Label text must follow the LABEL keyword

Cause: Missing label text about the LABEL keyword in the COMPUTE command.

Action: Check the syntax of the COMPUTE command in Chapter 5, "Command Reference" for the correct options.

SP2-0594 Usage: SET COLSEP [" \" *text*"]

Cause: Usage for SET COLSEP command.

Action: Specify a valid option.

SP2-0596 Usage: SET AUTO[COMMIT] [OFF | ON | IMM[EDIATE]]

Cause: An invalid option was used in the SET AUTO[COMMIT].

Action: Check the syntax of the SET AUTOCOMMIT command in Chapter 5, "Command Reference" for the correct options.

SP2-0597 *datatype_name* is not a valid *datatype_name* format

Cause: The value entered in the ACCEPT command was not in the specified datatype.

Action: Correct the datatype and re-enter.

SP2-0598 *value_name* does not match input format "*format_name*"

Cause: The value entered in the ACCEPT command was not in the specified format.

Action: Correct the format and try again.

SP2-0599 Usage: SET EDITF[ILE] *filename*.[*ext*]

Cause: Required filename was missing after the SET EDITFILE command.

Action: Check the syntax of the SET EDITFILE command in Chapter 5, "Command Reference" for the correct options.

SP2-0603 Usage: Illegal STORE command.

STORE [SET] *filename*.[*ext*] [CRE[ATE] | REP[LACE] | APP[END]]

Cause: An invalid STORE option was specified. Valid command clauses are CREATE, REPLACE or APPEND.

Action: Check the syntax of the STORE command in Chapter 5, "Command Reference" for the correct options.

SP2-0605 File *file_name* already exists. Use another name or STORE [SET] *filename*.[*ext*] REPLACE

Cause: The file specified in the STORE command already exists.

Action: Use the REPLACE option to overwrite the existing file, otherwise, specify another file name.

SP2-0606 Cannot create file *file_name*

Cause: The STORE command was unable to create the specified file. There may be insufficient disk space, too many open files, or read-only protection on the output directory.

Action: Check that there is enough disk space and that the protection on the directory allows creating a file.

SP2-0607 Cannot close file *file_name*

Cause: The STORE command was unable to close the specified file. Another resource may have locked the file.

Action: Check that the file is not locked before closing it.

SP2-0608 Object *object_name* is a remote object, cannot further describe

Cause: Unable to DESCRIBE the remote object.

Action: No action required.

SP2-0609 Usage: SET AUTOT[RACE] [OFF | ON | TRACE[ONLY]] [EXP[LAIN]] [STAT[ISTICS]]

Cause: An invalid option was used in the SET AUTOTRACE command.

Action: Check the syntax of the SET AUTOTRACE command in Chapter 5, "Command Reference" for the correct options.

SP2-0610 Error initializing *feature_name*

Cause: Not enough memory to enable this feature.

Action: Free up additional memory by closing applications not required, or reduce the size of the command, statement or query output.

SP2-0612 Error generating *report_name* report

Cause: Unable to generate the report using AUTOTRACE.

Action: Make a note of the message and the number, then contact the Database Administrator.

SP2-0613 Unable to verify PLAN_TABLE format or existence

Cause: An AUTOTRACE command was issued when the current user did not have the appropriate privilege to execute it.

Action: Make sure AUTOTRACE has the privileges and the objects to run. Make a note of the message and the number, then contact the Database Administrator.

SP2-0614 Server version too low for this feature

Cause: The current version of the Oracle Server is too low for this feature.

Action: Use a higher version of the Oracle Server.

SP2-0617 Cannot construct a unique STATEMENT_ID

Cause: Unable to construct a unique statement ID in AUTOTRACE.

Action: Check that AUTOTRACE is configured and that you have the PLUS-TRACE role enabled.

SP2-0618 Cannot find the Session Identifier. Check PLUSTRACE role is enabled

Cause: Unable to find the session identifier.

Action: Check and make sure that the PLUSTRACE role is enabled.

SP2-0619 Error while connecting

Cause: An error occurred while AUTOTRACE attempted to make a second connection to the database instance.

Action: Check that the database limit on number of active sessions has not been exceeded.

SP2-0620 Error while disconnecting

Cause: An error occurred while AUTOTRACE attempted to disconnect from the database instance.

Action: Check that the database is still available.

SP2-0621 Error ORA -*error_number* while gathering statistics

Cause: No data was found in the PLAN_TABLE while gathering statistics while using AUTOTRACE.

Action: Refer to the *Oracle9i Error Messages* for the specified ORA error message.

SP2-0622 Starting line number must be less than ending line number

Cause: The starting line number specified is larger than the ending number.

Action: Re-enter the starting line number with a smaller line number.

SP2-0623 Error accessing PRODUCT_USER_PROFILE.

Warning: Product user profile information not loaded!

Error in disabling roles in product user profile

Cause: These error messages are warnings that the PRODUCT_USER_PROFILE table has not been built in the SYSTEM account.

Action: The exact format of the file extension and location of the file are system dependent. See the SQL*Plus installation guide provided for your operating system. The script must be run as user SYSTEM.

SP2-0625 Error printing variable *variable_name*

Cause: Error encountered while printing the specified variable.

Action: Check and make sure that the specified variable is correct and try again.

SP2-0626 Error accessing package DBMS_APPLICATION_INFO

Cause: This message is followed by a successful login to the Oracle Server. The DBMS_APPLICATION package is used to maintain on-line information about a particular application logged onto Oracle. SET APPINFO could not be initialized.

Action: This package is created during the running of the CATPROC.SQL and should be available on all databases from Oracle 7.2. Check that your database is correctly installed.

**SP2-0631 String beginning *string_name* is too long.
Maximum size is *string_length* characters**

Cause: The string specified was too long.

Action: Reduce the size of the specified string and re-try the operation.

**SP2-0640 Not connected.
PASSW[ORD] [*username*]**

Cause: The PASSWORD command was issued when there was no connection to the Oracle instance

Action: Connect to the Oracle instance and re-try the operation; connect to the database before re-issuing the PASSWORD command.

SP2-0641 *command_name* requires connection to server

Cause: SQL*Plus was unable to execute the command because there was no connection to a database.

Action: Connect to a database and re-try the operation.

**SP2-0642 SQL*Plus internal error state *error_state* context *error_number*.
Unsafe to proceed**

Cause: An internal error occurred.

Action: Make a note of the message and the numbers, then contact Oracle Support Services.

**SP2-0645 Operating System error occurred
Unable to complete EDIT command**

Cause: An operating system error occurred with the EDIT command.

Action: Check that the file was created successfully, and verify that the device you are writing to is still available.

SP2-0650 New passwords do not match

Cause: The new passwords entered did not match.

Action: Re-issue the PASSWORD command and make sure the same new passwords are entered correctly.

SP2-0659 Password unchanged

Cause: The PASSWORD command failed to change passwords because:

- No passwords were given.
- The new passwords did not match.

Action: Re-issue the PASSWORD command and make sure that the new passwords are entered correctly.

SP2-0666 WARNING: SHIFTINOUT only affects shift sensitive character sets

Cause: The NLS character set used in this session does not contain shift sensitive characters. The SET SHIFTINOUT command is unnecessary.

Action: No action required.

SP2-0667 Message file *facility<lang>.msb* not found

Cause: The SP1, SP2, or CPY message file could not be found. SQL*Plus cannot run.

Action: Check the Oracle platform specific documentation to make sure SQL*Plus is installed correctly. This may occur because the ORACLE_HOME environment variable or registry equivalent is not set to the location of the Oracle software. Make sure this value is set correctly. Check that the SQL*Plus binary message files exist in the SQL*Plus message directory, for example \$ORACLE_HOME/sqlplus/mesg. Check the value of NLS_LANG environment variable or registry equivalent is correct.

SP2-0668 Invalid variable name

Cause: An invalid character was specified as part of the name.

Action: Specify the variable with valid characters.

SP2-0669 Valid characters are alphanumeric and ' _ '

Cause: An invalid character was specified as part of the name.

Action: Specify the variable with alphanumeric characters and ' _ '.

SP2-0670 Internal number conversion failed

Cause: A conversion request could not be performed because the string contained alphanumeric characters.

Action: Make sure that the string only contain numeric digits.

SP2-0675 COPY command not available

Cause: The COPY command is not available in this version of SQL*Plus.

Action: Make a note of the message and the number, then contact Oracle Support Services.

SP2-0676 Bind variable length cannot exceed *variable_length_units_of_variable*

Cause: The length of the bind variable datatype was exceeded.

Action: Reduce the length of the bind variable datatype.

SP2-0678 Column or attribute type can not be displayed by SQL*Plus

Cause: The type specified is not supported.

Action: Rewrite the query to select the data with types that SQL*Plus supports.

SP2-0685 The date *entered_variable* is invalid or format mismatched *format*

Cause: An invalid date was entered or does not match the format.

Action: Enter a valid date or date format.

SP2-0686 Usage: DESCRIBE [*schema.*]object[@*db_link*]

Cause: Usage for the DESCRIBE command.

Action: Check the syntax of the DESCRIBE command in Chapter 5, "Command Reference" for the correct options.

SP2-0691 Expected SYSDBA or SYSOPER, not *command_name*

Cause: Attempted to use the CONNECT AS syntax and specified something other than SYSDBA or SYSOPER.

Action: Correct the syntax and issue the CONNECT command again.

SP2-0692 Usage: CONN[ECT] [*login*] [AS [SYSDBA | SYSOPER]]

Where <*login*> ::= <*username*>[/<*password*>][@<*connect_string*>] | /

Cause: Usage for SQL*Plus CONNECT command.

Action: Check the syntax for the CONNECT command in Chapter 5, "Command Reference" for the correct usage.

SP2-0714 Invalid combination of STARTUP options

Cause: The specified options of the STARTUP command cannot be used simultaneously.

Action: Check the syntax of the STARTUP command in Chapter 5, "Command Reference" for the correct usage.

SP2-0715 Invalid combination of SHUTDOWN options

Cause: The specified options of the SHUTDOWN command cannot be used simultaneously.

Action: Check the syntax of the SHUTDOWN command in Chapter 5, "Command Reference" for the correct usage.

SP2-0716 Invalid combination of ARCHIVE LOG options

Cause: The specified options of the ARCHIVE LOG command cannot be used simultaneously.

Action: Check the syntax of the ARCHIVE LOG command in Chapter 5, "Command Reference" for the correct usage.

SP2-0717 Illegal SHUTDOWN option

Cause: An invalid option was used in the SHUTDOWN command.

Action: Check the syntax of the SHUTDOWN command in Chapter 5, "Command Reference" for the correct options.

SP2-0718 Illegal ARCHIVE LOG option

Cause: An invalid option was used in the ARCHIVE LOG command.

Action: Check the syntax of the ARCHIVE LOG command in Chapter 5, "Command Reference" for the correct options.

SP2-0728 Specify log: [<RET>=suggested | *filename* | AUTO | CANCEL]

Cause: This is a RECOVER DATABASE command prompt, prompting for the redo log files to be applied.

Action: Enter one of the above options.

SP2-0729 Cannot SET INSTANCE while connected to a database

Cause: There is a problem with the connection instance while issuing the SET INSTANCE command.

Action: Disconnect from the instance before re-issuing the command.

SP2-0733 Invalid connect string

Cause: Invalid connect string was specified.

Action: Check and make sure that connect string is correct.

SP2-0734 Unknown command beginning *command_name* ... - rest of line ignored

Cause: The command entered was invalid.

Action: Check syntax and re-enter.

SP2-0735 Unknown *command_name* option beginning *option_name*

Cause: An invalid option was specified for a given command.

Action: Check the syntax in Chapter 5, "Command Reference" for the correct options of the command you used.

SP2-0736 Command line overflow while substituting into line beginning *string_name*

Cause: The maximum length of the command line was exceeded.

Action: Reduce the length of the data in the substitution variables used in the command.

SP2-0737 Usage: SET DESCRIBE [DEPTH [1 | n | ALL]] [LINENUM [ON | OFF]] [INDENT [ON | OFF]]

Cause: Usage message for SET DESCRIBE command.

Action: Check the syntax of the SET DESCRIBE command in Chapter 5, "Command Reference" for the correct options.

SP2-0738 Restricted command *command_name* not available

Cause: The command was restricted by the -RESTRICT command-line option for security reasons.

Action: Ask your systems administrator why SQL*Plus should be run with a "-RESTRICT" option.

SP2-0745 Usage: SET SQLPLUSCOMPAT[IBILITY] *version.release*.*[update]*

Cause: An invalid option was used in the SET SQLPLUSCOMPAT[IBILITY] command.

Action: Check the syntax of the SET SQLPLUSCOMPATIBILITY command in Chapter 5, "Command Reference" for the correct options.

SP2-0746 *command_option* option out of range (*lower* through *upper*)

Cause: The specified value was not in the range.

Action: Specify a value in the range.

SP2-0747 PAGESIZE must be at least *max_page_size* to run this query with LINESIZE *line_size*

Cause: The PAGESIZE setting was too small to display the specified LINESIZE.

Action: Increase the PAGESIZE to at least match the specified LINESIZE.

iSQL*Plus Error Messages

SP2-0850 Command *command_name* is not available in iSQL*Plus

Cause: The command was not recognized or it is disabled in iSQL*Plus. Some commands are disabled because they are not relevant in a browser context or because they pose a security risk over the web.

Action: Remove the command from the script.

SP2-0851 Command beginning *command_name...* is not available in iSQL*Plus

Cause: The command was not recognized or it is disabled in iSQL*Plus. Some commands are disabled because they are not relevant in a browser context or because they pose a security risk over the web.

Action: Remove the command from the script.

SP2-0852 Option not available in iSQL*Plus

Cause: The command option is not available in iSQL*Plus. Some commands are disabled because they are not relevant in a browser context or because they pose a security risk over the web.

Action: Remove the option from the command.

SP2-0853 Empty username field

Cause: The username field was empty.

Action: Enter a username before attempting to log on.

SP2-0854 Password parsing error, please log in

Cause: An error occurred while parsing the username and password fields. iSQL*Plus cannot determine what was the intended password.

Action: Specify a password as part of the username or password in the "Password" field of the login screen. The password should appear once only.

SP2-0855 Connect identifier parsing error

Cause: An error occurred trying to read the connection identifier.

Action: If the full connection syntax is used in the username field (for example "username/password@connect_identifier") then the "Connect Identifier" field must be empty.

SP2-0856 Invalid username/password

Cause: iSQL*Plus was unable to connect to an Oracle instance; or the username and password were not correct.

Action: Make sure the Connect Identifiers are correct.

SP2-0857 iSQL*Plus restarted

Cause: Either the Oracle HTTP server handling the request has timed out and the session has expired, or the web server has halted. In either case session information is not available.

Action: Restart the Oracle HTTP server (if necessary) and re-login to iSQL*Plus.

SP2-0858 Usage: SET MARKUP HTML [HEAD *text*] [BODY *text*] [TABLE *text*] [ENTMAP [ON | OFF]] [PRE[FORMAT] [ON | OFF]]

Cause: An option to SET MARKUP is not valid in iSQL*Plus.

Action: Remove the invalid option.

SP2-0860 For a list of known commands enter HELP

Cause: Invalid command sequence entered.

Action: Use the HELP command to show the syntax of SQL*Plus commands.

SP2-0864 Session has expired. Please log in again

Cause: The iSQL*Plus session was idle for too long and the context has been removed to free resources for other connections.

Action: Reconnect to iSQL*Plus. The system administrator can configure the timeout period.

SP2-0866 Please enter statements in the input pane

Cause: Clicked the "Execute" button without any statements in the input pane to execute.

Action: Enter the statements to run in the input pane and click "Execute" again.

SP2-0867 No script to be saved

Cause: Clicked "Save" Script without having any statements in the input pane to save.

Action: Enter the statements to save into the input pane and click Save Script again.

SP2-0868 No output to be saved

Cause: The report did not generate any output.

Action: Make sure the script generates some results.

SP2-0869 Invalid file content

Cause: Attempted to load a script into the input area with a format the web server cannot understand.

Action: Make sure that the script is in a text file and the MIME type settings needed by the browser to recognize the file are set correctly. Typically if you are loading a file with the extension .SQL, make sure the browser has a "SQL" MIME type.

SP2-0871 No script to load

Cause: Clicked "Load Script" but did not have a file name specified.

Action: Enter a file name into the field and try again.

SP2-0872 SET AUTORECOVERY ON must be used in iSQL*Plus

Cause: Attempted to use the interactive mode of the RECOVER command in iSQL*Plus. Use the autorecovery mode in iSQL*Plus.

Action: Switch SET AUTORECOVERY ON and rerun the RECOVER command.

SP2-0873 An unexpected quote was found in the URI argument

Cause: A quote was found in the middle of the value portion of a keyword/value pair.

Action: Check for and remove the extra quote.

SP2-0874 URI argument is missing a keyword

Cause: There is no keyword in a keyword/value pair of a URI argument.

Action: Check for missing keyword or a missing equals sign.

SP2-0875 URI argument contains a keyword but no value"

Cause: A keyword was found but its associated value is missing in a keyword/value pair of a URI.

Action: Check for missing keyword or perhaps a missing equals sign.

SP2-0876 URI argument is missing an end quote

Cause: Could not find the end quote to match an open quote in a keyword/value pair of a URI.

Action: Check for and insert matching end quote.

SP2-0877 Found an unexpected character in a URI argument

Cause: Already have a keyword/value pair where the value is quoted but extra characters were still found in a URI argument.

Action: Remove the extra characters.

SP2-0878 Duplicate keyword "%s" specified

Cause: The keyword has already been specified in a previous parameter of a URI argument.

Action: Remove one of the keyword occurrences.

SP2-0879 Must specify a script for dynamic reports to execute

Cause: The script keyword was not specified in the URI.

Action: Add a script keyword and value to the URI argument.

SP2-0880 Enter connection details to run script

%s

Cause: No connection string was supplied in the URI argument for a dynamic report. That is, the userid keyword was not specified.

Action: Login via the interactive login page.

COPY Command Messages

CPY0002 Illegal or missing APPEND, CREATE, INSERT, or REPLACE option

Cause: An internal COPY function has invoked COPY with a create option (flag) value that is out of range.

Action: Please contact Oracle Worldwide Customer Support Services.

CPY0003 Internal Error: logical host number out of range

Cause: An internal COPY function has been invoked with a logical host number value that is out of range.

Action: Please contact Oracle Worldwide Customer Support Services.

CPY0004 Source and destination table and column names don't match

Cause: On an APPEND operation or an INSERT (when the table exists), at least one column name in the destination table does not match the corresponding column name in the optional column name list or in the SELECT command.

Action: Re-specify the COPY command, making sure that the column names and their respective order in the destination table match the column names and column order in the optional column list or in the SELECT command

CPY0005 Source and destination column attributes don't match

Cause: On an APPEND operation or an INSERT (when the table exists), at least one column in the destination table does not have the same datatype as the corresponding column in the SELECT command.

Action: Re-specify the COPY command, making sure that the data types for items being selected agree with the destination. Use TO_DATE, TO_CHAR, and TO_NUMBER to make conversions.

CPY0006 Select list has more columns than destination table

Cause: On an APPEND operation or an INSERT (when the table exists), the number of columns in the SELECT command is greater than the number of columns in the destination table.

Action: Re-specify the COPY command, making sure that the number of columns being selected agrees with the number in the destination table.

CPY0007 Select list has fewer columns than destination table

Cause: On an APPEND operation or INSERT (when the table exists), the number of columns in the SELECT command is less than the number of columns in the destination table.

Action: Re-specify the COPY command, making sure that the number of columns being selected agrees with the number in the destination table.

CPY0008 More column list name than columns in the destination table

Cause: On an APPEND operation or an INSERT (when the table exists), the number of columns in the column name list is greater than the number of columns in the destination table.

Action: Re-specify the COPY command, making sure that the number of columns in the column list agrees with the number in the destination table.

CPY0009 Fewer column list name than columns in the destination table

Cause: On an APPEND operation or an INSERT (when the table exists), the number of columns in the column name list is less than the number of columns in the destination table.

Action: Re-specify the COPY command, making sure that the number of columns in the column list agrees with the number in the destination table.

B

Security

This appendix describes the available methods for controlling access to database tables and SQL*Plus commands. It covers the following topics:

- PRODUCT_USER_PROFILE Table
- Disabling SQL*Plus, SQL, and PL/SQL Commands
- Creating and Controlling Roles

PRODUCT_USER_PROFILE Table

Various Oracle products use the PRODUCT_USER_PROFILE (PUP) table, a table in the SYSTEM account, to provide product-level security that supplements the user-level security provided by the SQL GRANT and REVOKE commands and user roles.

DBAs can use the PUP table to disable certain SQL and SQL*Plus commands in the SQL*Plus environment on a per-user basis. SQL*Plus—not Oracle—enforces this security. DBAs can even restrict access to the GRANT, REVOKE, and SET ROLE commands to control users' ability to change their database privileges.

SQL*Plus reads restrictions from the PUP table when a user logs in to SQL*Plus and maintains those restrictions for the duration of the session. Changes to the PUP table will only take effect the next time the affected users log in to SQL*Plus.

When SYSTEM, SYS, or a user authenticating with AS SYSDBA or AS SYSOPER privileges connects or logs in, SQL*Plus does not read the PUP table. Therefore, no restrictions apply to these users.

The PUP table applies only to the local database. If accessing objects on a remote database via a database link, the PUP table for the remote database does not apply. The remote database cannot extract the username and password from the database link in order to determine that user's profile and privileges.

Creating the PUP Table

You can create the PUP table by running the command file named PUPBLD with the extension SQL as SYSTEM. The exact format of the file extension and the location of the file are system dependent. See the Oracle installation and user's manual(s) provided for your operating system or your DBA for more information.

Note: If the table is created incorrectly, all users other than privileged users will see a warning when connecting to Oracle that the PUP table information is not loaded.

PUP Table Structure

The PUP table has the following columns:

PRODUCT	NOT NULL CHAR (30)
USERID	CHAR (30)
ATTRIBUTE	CHAR (240)
SCOPE	CHAR (240)
NUMERIC_VALUE	NUMBER (15,2)
CHAR_VALUE	CHAR (240)
DATE_VALUE	DATE
LONG_VALUE	LONG

Description and Use of PUP Columns

Refer to the following list for the descriptions and use of each column in the PUP table:

Product	Must contain the product name (in this case "SQL*PLUS"). You cannot enter wildcards or NULL in this column.
userid	Must contain the username (in uppercase) of the user for whom you wish to disable the command. To disable the command for more than one user, use SQL wild cards (%) or make multiple entries. Thus, all of the following entries are valid: <ul style="list-style-type: none"> ■ HR ■ CLASS1 ■ CLASS% (all users whose names start with CLASS) ■ % (all users)
Attribute	Must contain the name (in uppercase) of the SQL, SQL*Plus, or PL/SQL command to disable (for example, RUN). If you are disabling a role, it must contain the character string "ROLES". You cannot enter a wildcard. See the section "PUP Table Administration" later in this chapter for a list of SQL and SQL*Plus commands you can disable.
Scope	SQL*Plus ignores this column. It is recommended that you enter NULL in this column. Other products may store specific file restrictions or other data in this column.

Numeric_Value	SQL*Plus ignores this column. It is recommended that you enter NULL in this column. Other products may store numeric values in this column.
Char_Value	Must contain the character string "DISABLED" to disable a SQL, SQL*Plus, or PL/SQL command. If you are disabling a role, it must contain the name of the role you wish to disable. You cannot use a wildcard.
Date_Value	SQL*Plus ignores this column. It is recommended that you enter NULL in this column. Other products may store DATE values in this column.
Long_Value	SQL*Plus ignores this column. It is recommended that you enter NULL in this column. Other products may store LONG values in this column.

PUP Table Administration

The DBA username SYSTEM owns and has all privileges on the PUP table. Other Oracle usernames should have only SELECT access to this table, which allows a view of restrictions of that username and those restrictions assigned to PUBLIC. The command file PUPBLD.SQL, when run, grants SELECT access on the PUP table to PUBLIC.

Disabling SQL*Plus, SQL, and PL/SQL Commands

To disable a SQL or SQL*Plus command for a given user, insert a row containing the user's username in the Userid column, the command name in the Attribute column, and DISABLED in the Char_Value column.

The Scope, Numeric_Value, and Date_Value columns should contain NULL. For example:

PRODUCT	USERID	ATTRIBUTE	SCOPE	NUMERIC VALUE	CHAR VALUE	DATE VALUE
-----	-----	-----	-----	-----	-----	-----
SQL*Plus	HR	HOST			DISABLED	
SQL*Plus	%	INSERT			DISABLED	
SQL*Plus	%	UPDATE			DISABLED	
SQL*Plus	%	DELETE			DISABLED	

To re-enable commands, delete the row containing the restriction.

You can disable the following SQL*Plus commands:

COPY	SET (see note below)
EXECUTE	START
RUN	

Note: Disabling the SQL*Plus SET command will also disable the SQL SET ROLE and SET TRANSACTION commands. Disabling the SQL*Plus START command will also disable the SQL*Plus @ and @@ commands.

You can also disable the following SQL commands:

ALTER	LOCK
ANALYZE	NOAUDIT
AUDIT	RENAME
CONNECT	REVOKE
CREATE	SELECT
DELETE	SET ROLE
DROP	SET TRANSACTION
GRANT	TRUNCATE
INSERT	UPDATE

You can also disable the following PL/SQL commands:

BEGIN	DECLARE
-------	---------

Note: Disabling BEGIN and DECLARE does not prevent the use of the SQL*Plus EXECUTE command. EXECUTE must be disabled separately.

Example A–1 Setting Restrictions in the PUP Table

This is an example of how to insert a row into the PUP table to restrict the user HR from using the SELECT statement:

1. Log in with AS SYSDBA privileges.
2. Insert a row into the PUP table with the command:



```
INSERT INTO PRODUCT_USER_PROFILE
VALUES ('SQL*PLUS', 'HR', 'SELECT', NULL, NULL, 'DISABLED', NULL, NULL);
```

3. Connect as HR/HR and try to SELECT something:



```
CONNECT HR/HR;
SELECT * FROM EMP_DETAILS_VIEW;
```

This command causes the following error message:



```
SP2-0544: INVALID COMMAND: SELECT
```

4. To delete this row and remove the restriction from the user HR, CONNECT again with AS SYSDBA privileges and enter:



```
DELETE FROM PRODUCT_USER_PROFILE WHERE USERID = 'HR';
```

Creating and Controlling Roles

You can use SQL commands to create and control access to roles to provide security for your database tables.

By creating a role and then controlling who has access to it, you can ensure that only certain users have access to particular database privileges.

Roles are created and used with the SQL CREATE, GRANT, and SET commands:

- To create a role, you use the CREATE command. You can create roles with or without passwords.
- To grant access to roles, you use the GRANT command. In this way, you can control who has access to the privileges associated with the role.
- To access roles, you use the SET ROLE command. If you created the role with a password, the user must know the password in order to access the role.

For more information about roles, see your *Oracle9i SQL Reference*, your *Oracle9i Administrator's Guide*, and your *Oracle9i Concepts* manual.

Disabling SET ROLE

From SQL*Plus, users can submit any SQL command. In certain situations, this can cause security problems. Unless you take proper precautions, a user could use SET ROLE to access privileges obtained via an application role. With these privileges, they might issue SQL statements from SQL*Plus that could wrongly change database tables.

To prevent application users from accessing application roles in SQL*Plus, you can use the PUP table to disable the SET ROLE command. You also need to disable the BEGIN and SQL*Plus EXECUTE commands to prevent application users setting application roles through a PL/SQL block. This allows a SQL*Plus user only those privileges associated with the roles enabled when they started SQL*Plus. For more information about the creation and usage of user roles, see your *Oracle9i SQL Reference* and *Oracle9i Administrator's Guide*.

Disabling User Roles

To disable a role for a given user, insert a row in the PUP table containing the user's username in the Userid column, "ROLES" in the Attribute column, and the role name in the Char_Value column.

Note: When you enter "PUBLIC" or "%" for the Userid column, you disable the role for all users. You should only use "%" or "PUBLIC" for roles which are granted to "PUBLIC". If you try to disable a role that has not been granted to a user, none of the roles for that user are disabled.

The Scope, Numeric_Value, and Date_Value columns should contain NULL. For example:

PRODUCT	USERID	ATTRIBUTE	SCOPE	NUMERIC VALUE	CHAR VALUE	DATE VALUE
-----	-----	-----	-----	-----	-----	-----
SQL*Plus	HR	ROLES			ROLE1	
SQL*Plus	PUBLIC	ROLES			ROLE2	

During login, these table rows are translated into the command

```
SET ROLE ALL EXCEPT ROLE1, ROLE2
```

To ensure that the user does not use the `SET ROLE` command to change their roles after login, you can disable the `SET ROLE` command. See "Disabling `SET ROLE`" earlier in this appendix.

To re-enable roles, delete the row containing the restriction.

Unsupported SQL*Plus Commands

This appendix lists SQL*Plus commands not supported in this release of *iSQL*Plus*. Attempting to use any of the following unsupported commands or command options raises an SP2-0850 error message.

The ACCEPT, CLEAR SCREEN, PASSWORD and PAUSE commands, and the following SET command options have no context in *iSQL*Plus* and have not been implemented.

COLSEP	SQLPREFIX
EDITFILE	SQLPROMPT
FLUSH	SUFFIX
NEWPAGE	TAB
PAUSE	TERMOUT
SHIFTINOUT	TIME
SQLBLANKLINES	TRIMOUT
SQLCONTINUE	TRIMSPool
SQLNUMBER	

The following commands have security issues on the middle tier and have not been implemented.

EXIT/QUIT	SAVE	WHENEVER OSERROR EXIT
GET	SPOOL	WHENEVER SQLError EXIT
HOST	STORE	

The following commands are SQL buffer editing commands which are not relevant in *iSQL*Plus* and have not been implemented.

[0-9]+	CHANGE	EDIT
APPEND	DEL	INPUT

Glossary

account

An authorized user of an operating system or a product (such as Oracle database server or Oracle Forms). Depending on the operating system, may be referred to as ID, User ID, login, and so on. Accounts are often created and controlled by a system administrator.

alias

In SQL, a temporary name assigned to a table, view, column, or value within a SQL statement, used to refer to that item later in the same statement or in associated SQL*Plus commands.

alignment

The way in which data is positioned in a field. It may be positioned to the left, right, center, flush/left, flush/right, or flush/center of the defined width of a field.

anonymous block

A PL/SQL program unit that has no name and does not require the explicit presence of the BEGIN and END keywords to enclose the executable statements.

archived redo log

Recovery structure where online redo log files are archived before being reused.

ARCHIVELOG

Redo log mode where the filled online redo log files are archived before they are reused in the cycle. In ARCHIVELOG mode, the database can be completely recovered from both instance and disk failure. The database can also be backed up while it is open and available for use. However, additional administrative

operations are required to maintain the archived redo log. See also archived redo log.

argument

A data item following the command file name in a START command. The argument supplies a value for a parameter in the command file.

array processing

Processing performed on multiple rows of data rather than one row at a time. In some Oracle utilities such as SQL*Plus, Export/Import, and the precompilers, users can set the size of the array; increasing the array size often improves performance.

ASCII

A convention for using digital data to represent printable characters. ASCII is an acronym for American Standard Code for Information Interchange.

autocommit

A feature unique to SQL*Plus that enables SQL*Plus to automatically commit changes to the database after every successful execution of a SQL command or PL/SQL block. Setting the AUTOCOMMIT variable of the SET command to ON enables this feature. Setting the AUTOCOMMIT variable to *n* enables this feature after every *n* successful INSERT, UPDATE or DELETE commands or PL/SQL blocks.

background process

A non-interactive process that runs in an operating system environment and performs some service or action. Certain Oracle database server products use background processes for different tasks, such as performing and coordinating tasks on behalf of concurrent users of the database, processing and delivering electronic messages, and managing printing services.

bind reference

A reference to a parameter used to replace a single literal value (for example, a character string, number, or date) appearing anywhere in a PL/SQL construct or a SQL SELECT statement. For a bind reference, you must precede the parameter name with a colon (:).

bind variable

A variable in a SQL statement that must be replaced with a valid value, or the address of a value, in order for the statement to successfully execute.

bit

The smallest unit of data. A bit only has two possible values, 0 or 1. Bits can be combined into groups of eight called bytes; each byte represents a single character of data. See also byte.

block

In PL/SQL, a group of SQL and PL/SQL commands related to each other through procedural logic.

body

A report region that contains the bulk of the report (text, graphics, data, and computations).

break

An event, such as a change in the value of an expression, that occurs while SQL*Plus processes a query or report. You can direct SQL*Plus to perform various operations, such as printing subtotals, whenever specified breaks occur.

break column

A column in a report that causes a break when its value changes and for which the user has defined break operations.

break group

A group containing one or more break columns.

break hierarchy

The order in which SQL*Plus checks for the occurrence of breaks and triggers the corresponding break operations.

break order

Indicates the order in which to display a break column's data. Valid options are Ascending and Descending.

break report

A report that divides rows of a table into "sets", based on a common value in the break column.

buffer

An area where the user's SQL statements or PL/SQL blocks are temporarily stored. The SQL buffer is the default buffer. You can edit or execute commands from multiple buffers; however, SQL*Plus does not require the use of multiple buffers.

byte

A group of eight sequential bits that represents a letter, number, or symbol (that is, a character). Treated as a unit of data by a computer.

CGI script

See Common Gateway Interface.

CHAR datatype

An Oracle datatype provided for ANSI/ISO compatibility. A CHAR column is a fixed-length column and can contain any printable characters, such as A, 3, &, or blanks, and can have from 1 to 2000 bytes or can be null. For more information about the CHAR datatype, refer to the *Oracle9i SQL Reference*.

character

A single location on a computer system capable of holding one alphabetic character or numeric digit. One or more characters are held in a field. One or more fields make up a record, and one or more records may be held in a file or database table.

character string

A group of sequential letters, numerals, or symbols, usually comprising a word or name, or portion thereof.

clause

A part of a SQL statement that does not constitute the full statement; for example, a "WHERE clause".

client

A user, software application, or computer that requests the services, data, or processing of another application or computer (the "server"). In a two-task environment, the client is the user process. In a network environment, the client is the local user process and the server may be local or remote.

CLOB datatype

A standard Oracle datatype. The CLOB datatype is used to store single-byte character large object data, and can store up to 4 gigabytes of character data.

column

A vertical space in a database table that represents a particular domain of data. A column has a column name and a specific datatype. For example, in a table of employee information, all of the employees' dates of hire would constitute one column. A record group column represents a database column.

column expression

An expression in a SELECT statement that defines which database column(s) are retrieved. It may be a column name or a valid SQL expression referencing a column name.

column heading

A heading created for each column appearing in a report.

command

An instruction to or request of a program, application, operating system, or other software, to perform a particular task. Commands may be single words or may require additional phrases, variously called arguments, options, parameters, and qualifiers. Unlike statements, commands execute as soon as you enter them. ACCEPT, CLEAR, and COPY are examples of commands in SQL*Plus.

command file

A file containing a sequence of commands that you can otherwise enter interactively. The file is saved for convenience and re-execution. Command files are often called by operating-system specific names. In SQL*Plus, you can execute the command file with the START, @ or @@ commands.

command line

A line on a computer display on which typed in commands appear. An example of a command line is the area next to the DOS prompt on a personal computer. See also prompt.

command prompt

The text, by default SQL>, with which SQL*Plus requests your next command.

comment

A language construct for the inclusion of explanatory text in a program, the execution of which remains unaffected.

commit

To make permanent changes to data (inserts, updates, deletes) in the database. Before changes are committed, both the old and new data exist so that changes can be stored or the data can be restored to its prior state.

Common Gateway Interface

The Common Gateway Interface (CGI) describes a part of a web server that allows user interaction, typically via a web browser, with programs running on the server. CGI scripts enable this user interaction to create dynamic web pages or web page elements, or to take user input and respond accordingly. A very common use is to provide an interactive form which a user completes online and then submits. Some common languages in use for CGI scripts are Perl, JavaScript and Java

computation

Used to perform runtime calculations on data fetched from the database. These calculations are a superset of the kinds of calculations that can be done directly with a SELECT statement. See also formula column.

computed column

See computation.

configuration

In Oracle Net, the set of instructions for preparing network communications, as outlined in the Oracle Net documentation.

configuration files

Files that are used to identify and characterize the components of a network. Configuration is largely a process of naming network components and identifying relationships among those components.

connect

To identify yourself to Oracle by entering your username and password in order to gain access to the database. In SQL*Plus, the CONNECT command allows you to log off Oracle and then log back on with a specified username.

connect identifier

The set of parameters, including a protocol, that Oracle Net uses to connect to a specific Oracle instance on the network.

current line

In an editor, such as the SQL*Plus editor, the line in the current buffer that editing commands will currently affect.

database

A set of operating system files, treated as a unit, in which an Oracle database server stores a set of data dictionary tables and user tables. A database requires three types of files: database files, redo log files, and control files.

database administrator (DBA)

(1) A person responsible for the operation and maintenance of an Oracle database server or a database application. The database administrator monitors its use in order to customize it to meet the needs of the local community of users. (2) An Oracle username that has been given DBA privileges and can perform database administration functions. Usually the two meanings coincide. There may be more than one DBA per site.

database instance failure

Failure that occurs when a problem arises that prevents an Oracle database instance (SGA and background processes) from continuing to work. Instance failure may result from a hardware problem such as power outage, or a software problem, such as operating system crash. When an instance failure occurs, the data in the buffers of the SGA is not written to the datafiles.

database link

An object stored in the local database that identifies a remote database, a communication path to the remote database, and optionally, a username and password for it. Once defined, a database link can be used to perform queries on tables in the remote database. Also called *DBlink*. In SQL*Plus, you can reference a database link in a DESCRIBE or COPY command.

database object

Something created and stored in a database. Tables, views, synonyms, indexes, sequences, clusters, and columns are all examples of database objects.

database server

The computer which runs the ORACLE Server kernel and contains the database.

database specification

An alphanumeric code that identifies a database, used to specify the database in Oracle Net operations and to define a database link. In SQL*Plus, you can reference a database specification in a COPY, CONNECT, or SQLPLUS command.

database string

A string of Oracle Net parameters used to indicate the network prefix, the host system you want to connect to, and the system ID of the database on the host system.

Data Control Language (DCL)

The category of SQL statements that control access to the data and to the database. Examples are the GRANT and REVOKE statements. Occasionally DCL statements are grouped with DML statements.

Data Definition Language (DDL)

The category of SQL statements that define or delete database objects such as tables or views. Examples are the CREATE, ALTER, and DROP statements.

data dictionary

A comprehensive set of tables and views automatically created and updated by the Oracle database server, which contains administrative information about users, data storage, and privileges. It is installed when Oracle is initially installed and is a central source of information for the Oracle database server itself and for all users of Oracle. The tables are automatically maintained by Oracle. It is sometimes referred to as the *catalog*.

Data Manipulation Language (DML)

The category of SQL statements that query and update the database data. Common DML statements are SELECT, INSERT, UPDATE, and DELETE. Occasionally DCL statements are grouped with DML statements.

data security

The mechanisms that control the access and use of the database at the object level. For example, data security includes access to a specific schema object and the specific types of actions allowed for each user on the object (for example, user HR can issue SELECT and INSERT statements but not DELETE statements using the EMP table). It also includes the actions, if any, that are audited for each schema object.

datatype

(1) A standard form of data. The Oracle datatypes are CHAR, NCHAR, VARCHAR2, NVARCHAR2, DATE, NUMBER, LONG, CLOB, NCLOB, RAW, and LONG RAW; however, the Oracle database server recognizes and converts other standard datatypes. (2) A named set of fixed attributes that can be associated with an item as a property. Data typing provides a way to define the behavior of data.

DATE datatype

A standard Oracle datatype used to store date and time data. Standard date format is DD-MMM-YY, as in 23-NOV-98. A DATE column may contain a date and time between January 1, 4712 BC to December 31, 9999 AD.

DBA

See database administrator (DBA).

DCL

See Data Control Language (DCL).

DDL

See Data Definition Language (DDL).

default

A clause or option value that SQL*Plus uses if you do not specify an alternative.

default database

See local database.

directory

On some operating systems, a named storage space for a group of files. It is actually one file that lists a set of files on a particular device.

dismounted database

A database that is not mounted by any instance, and thus cannot be opened and is not currently available for use.

display format

See format.

display width

The number of characters or spaces allowed to display the values for an output field.

DML

See Data Manipulation Language (DML).

DUAL table

A standard Oracle database table named DUAL, which contains exactly one row. The DUAL table is useful for applications that require a small “dummy” table (the data is irrelevant) to guarantee a known result, such as “true.”

editor

A program that creates or modifies files.

end user

The person for whom a system is being developed; for example, an airline reservations clerk is an end user of an airline reservations system. See also SQL*Plus.

error message

A message from a computer program (for example, SQL*Plus) informing you of a potential problem preventing program or command execution.

expression

A formula, such as SALARY + COMMISSION, used to calculate a new value from existing values. An expression can be made up of column names, functions, operators, and constants. Formulas are found in commands or SQL statements.

extension

On some operating systems, the second part of the full file specification. Several standard file extensions are used to indicate the type or purpose of the file, as in file extensions of SQL, LOG, LIS, EXE, BAT, and DIR. Called file type on some operating systems.

file

A collection of data treated as a unit, such as a list, document, index, note, set of procedures, and so on. Generally used to refer to data stored on magnetic tapes or disks. See also filename, extension, and file type.

filename

The name component of a file specification. A filename is assigned by either the user or the system when the file itself is created. See also extension and file type.

file type

On some operating systems, the part of the filename that usually denotes the use or purpose of the file. See extension.

format

Columns contain information in one of four types; users can specify how they want a query to format information it retrieves from character, number, date, or long columns. For example, they can choose to have information of type date appear as 23/11/98, or Monday Twenty-third November 1998, or any other valid date format.

format model

A clause element that controls the appearance of a value in a report column. You specify predefined format models in the COLUMN, TTITLE, and BTITLE commands' FORMAT clauses. You can also use format models for DATE columns in SQL date conversion functions, such as TO_DATE.

form feed

A control character that, when executed, causes the printer to skip to the top of a new sheet of paper (top of form). When SQL*Plus displays a form feed on most terminals, the form feed clears the screen.

formula column

Manually-created column that gets its data from a PL/SQL procedure, function, or expression, user exit, SQL statement, or any combination of these.

function

A PL/SQL subprogram that executes an operation and returns a value at the completion of the operation. A function can be either built-in or user-named. Contrast with procedure.

heading

In SQL*Plus, text that names an output column, appearing above the column. See also column heading.

host computer

The computer from which you run SQL*Plus.

HTML

HTML (HyperText Markup Language) is the language used to write most of the documents available on the World Wide Web. It provides display and linking specifications that are recognized by most web browsers. The HTML recommendation is sponsored by the World Wide Web Consortium (w3) and further details about the w3 and the HTML recommendation can be found at the w3 web site: <http://www.w3.org>.

instance

The background processes and memory area required to access an Oracle database. A database system requires one instance and one database. An Oracle database server consists of an SGA and a set of Oracle database server system processes.

instance failure

See database instance failure.

instance recovery

Recovery of an instance in the event of software or hardware failure, so that the database is again available to users. If the instance terminates abnormally, then the instance recovery automatically occurs at the next instance startup.

iSQL*Plus

iSQL*Plus is a browser-based interface to SQL*Plus. It is a component of the SQL*Plus product.

iSQL*Plus enables you to use a web browser to connect to Oracle9i and perform the same actions as you would through the command line version of SQL*Plus. You can use iSQL*Plus to write SQL*Plus, SQL and PL/SQL commands to:

- Enter, edit, run and save SQL commands and PL/SQL blocks.
- Calculate, and print query results.
- List column definitions for any table.
- Access and copy data between databases.
- Perform database administration.

iSQL*Plus Server

The iSQL*Plus Server is located on the middle tier of the iSQL*Plus three-tier architecture. It comprises the SQL*Plus engine and the iSQL*Plus module. In combination with the web server provided by the Oracle HTTP Server, they provide the iSQL*Plus web enabled interface to Oracle9i.

Julian date

An algorithm for expressing a date in integer form, using the SQL function JDATE. Julian dates allow additional arithmetic functions to be performed on dates.

justification

See alignment.

label

Defines the label to be printed for the computed value in the COMPUTE command. The maximum length of a COMPUTE label is 500 characters.

LGWR

See Log Writer (LGWR).

local database

The database that SQL*Plus connects to when you start SQL*Plus, ordinarily a database on your host computer. Also called a default database. See also remote database.

log in (or log on)

To perform a sequence of actions at a terminal that establishes a user's communication with the operating system and sets up default characteristics for the user's terminal session.

log off (or log out)

To terminate interactive communication with the operating system, and end a terminal session.

Log Writer (LGWR)

A background process used by an Oracle instance. LGWR writes redo log entries to disk. Redo log data is generated in the redo log buffer of the system global area. As transactions commit and the log buffer fills, LGWR writes redo log entries into an online redo log file.

logon string

A user-specified command line, used to run an application that is connected to either a local or remote database. The logon string either explicitly includes a connect identifier or implicitly uses a default connect identifier.

LONG datatype

One of the standard Oracle datatypes. A LONG column can contain any printable characters such as A, 3, &, or a blank, and can have any length from 0 to 2 gigabytes.

MARKUP

Refers to the SET MARKUP clause or the SQLPLUS -MARKUP clause that permits SQL*Plus output to be generated in HTML format for delivery on the Internet. SQL*Plus output generated in HTML can be viewed with any web browser supporting HTML 3.2.

mounted database

A database associated with an Oracle instance. The database may be opened or closed. A database must be both mounted and opened to be accessed by users. A database that has been mounted but not opened can be accessed by DBAs for some maintenance purposes.

multi-threaded server

Allows many user processes to share a small number of server processes, minimizing the number of server processes and maximizing the utilization of available system resources.

NCHAR datatype

Beginning with Oracle9i, the NCHAR datatype is redefined to be a Unicode-only datatype. The NCHAR datatype specifies a fixed-width national character set character string, with width specifications referring to the number of characters, and can have a maximum column size up to 2000 bytes. For more information about the NCHAR datatype, refer to the *Oracle9i SQL Reference*.

NCLOB datatype

A standard Oracle datatype. The NCLOB datatype is used to store fixed-width national character set character (NCHAR) data, and can store up to 4 gigabytes of character text data.

Net8

See Oracle Net.

network

A group of two or more computers linked together through hardware and software to allow the sharing of data and/or peripherals.

NLS_LENGTH_SEMANTICS

NLS_LENGTH_SEMANTICS is an environmental parameter used by the SQL*Plus client application to enable you to create CHAR and VARCHAR2 columns and variables using either byte or character length semantics. NCHAR, NVARCHAR2, CLOB and NCLOB columns are always character-based, and hence are not affected by this variable. If this variable has not been explicitly set at session startup, a default value of BYTE is used (Byte length semantics). The value of NLS_LENGTH_SEMANTICS is then applied as the length semantics of any CHAR or VARCHAR2 declarations which DO NOT explicitly state the length qualifier. NLS_LENGTH_SEMANTICS is also used when displaying variables, or describing tables, views, synonyms, or other objects. On the server side, NLS_LENGTH_SEMANTICS can be set as an initialization parameter, and can be dynamically altered via the 'ALTER SESSION' and 'ALTER SYSTEM' SQL commands. For more information about setting NLS_LENGTH_SEMANTICS on the server side, refer to the *Oracle9i Globalization Support Guide*. Note that NLS_LENGTH_SEMANTICS may differ between the client and server, but the issuing of an ALTER SESSION SET NLS_LENGTH_SEMANTICS=value command to alter the session scope, will be reflected in the SQL*Plus session.

null

A value that means, “a value is not applicable” or “the value is unknown”. Nulls are not equal to any specific value, even to each other. Comparisons with nulls are always false.

NULL value

The absence of a value.

NUMBER datatype

A standard Oracle datatype. A NUMBER column can contain a number, with or without a decimal point and a sign, and can have from 1 to 105 decimal digits (only 38 digits are significant).

NVARCHAR2 datatype

Beginning with Oracle9i, the NVARCHAR2 datatype is redefined to be a Unicode-only datatype. The NVARCHAR2 datatype specifies a variable-width national character set character string, with width specifications referring to the number of characters, and can have a maximum column size up to 4000 bytes. For more information about the NVARCHAR2 datatype, refer to the *Oracle9i SQL Reference*.

object

An object is an instance of an object type. In Oracle9i, objects can be persistent (i.e. stored in the database) or transient (i.e. PL/SQL or Oracle Call Interface™ (OCI) variables). See also object type.

object-relational model

A database model that combines the key aspects of the relational and object data models into a single system. Oracle9i is an object-relational database system.

object type

A user-defined type that models a structure and behavior of an object. Equivalent to the concept of a class in different programming languages. In Oracle9i, object types have public interfaces with attributes and methods. Object types are also known as abstract data types.

online redo log

(1) Redo log files that have not been archived, but are either available to the instance for recording database activity or are filled and waiting to be archived or reused. (2) A set of two or more online redo log files that record all committed changes made to the database.

open database

A database that has been mounted and opened by an instance and is available for access by users. If a database is open, users can access the information it contains. See also mounted database.

operating system

The system software that manages a computer's resources, performing basic tasks such as allocating memory and allowing computer components to communicate.

Oracle Net

Oracle's remote data access software that enables both client-server and server-server communications across any network. Oracle Net supports distributed processing and distributed database capability. Oracle Net runs over and interconnects many communications protocols. Oracle Net is backward compatible with Net8 and SQL*Net version 2.

Oracle Server

The relational database management system (RDBMS) sold by Oracle Corporation. Components of Oracle Server include the kernel and various utilities for use by DBAs and database users.

output

Results of a report after it is run. Output can be displayed on a screen, stored in a file, or printed on paper.

output file

File to which the computer transfers data.

packages

A method of encapsulating and storing related procedures, functions, and other package constructs together as a unit in the database. While packages provide the database administrator or application developer organizational benefits, they also offer increased functionality and database performance.

page

A screen of displayed data or a sheet of printed data in a report.

parameter

A substitution variable consisting of an ampersand followed by a numeral (&1, &2, and so on.). You use parameters in a command file and pass values into them through the arguments of the START command.

parameter file

A file used by Oracle9i Server to provide specific values and configuration settings to be used on database startup. For more information about the function of the parameter file, see the *Oracle9i Administrator's Guide*.

password

A secondary identification word (or string of alphanumeric characters) associated with a username. A password is used for data security and known only to its owner. Passwords are entered in conjunction with an operating system login ID, Oracle username, or account name in order to connect to an operating system or software application (such as the Oracle database). Whereas the username or ID is public, the secret password ensures that only the owner of the username can use that name, or access that data.

PL/SQL

The 3GL Oracle procedural language extension of SQL. PL/SQL combines the ease and flexibility of SQL with the procedural functionality of a structured programming language, such as IF...THEN, WHILE, and LOOP. Even when PL/SQL is not stored in the database, applications can send blocks of PL/SQL to the database rather than individual SQL statements, thereby reducing network traffic.

PL/SQL is interpreted and parsed at runtime, it does not need to be compiled.

procedure

A set of SQL and PL/SQL statements grouped together as an executable unit to perform a very specific task. Procedures and functions are nearly identical; the only difference between the two is that functions always return a single value to the caller, while procedures do not return a value to the caller.

process

(1) A thread of control in an operating system; that is, a mechanism in an operating system that can execute a series of steps. Some operating systems use the terms job or task. A process normally has its own private memory area in which it runs.

prompt

(1) A message from a computer program that instructs you to enter data or take some other action. (2) Word(s) used by the system as a cue to assist a user's response. Such messages generally ask the user to respond by typing some information in the adjacent field. See also command line.

query

A SQL SELECT statement that retrieves data, in any combination, expression, or order. Queries are read-only operations; they do not change any data, they only retrieve data. Queries are often considered to be DML statements.

query results

The data retrieved by a query.

RAW datatype

A standard Oracle datatype, a RAW data column may contain data in any form, including binary. You can use RAW columns for storing binary (non-character) data.

RDBMS (Relational Database Management System)

An Oracle7 (and earlier) term. Refers to the software used to create and maintain the system, as well as the actual data stored in the database. See also Relational Database Management System (RDBMS), Server and Oracle Server.

record

A synonym for row; one row of data in a database table, having values for one or more columns.

recover

The Oracle process of restoring all or part of a database from specified redo log files.

redo log

A sequential log of all changes made to the data. The redo log is written and used in the event of a failure that prevents changes from being written to disk. The redo log consists of two or more redo log files.

redo log file

A file containing records of all changes made to the database. These files are used for recovery purposes. See also redo log.

Relational Database Management System (RDBMS)

An Oracle7 (and earlier) term. A computer program designed to store and retrieve shared data. In a relational system, data is stored in tables consisting of one or more rows, each containing the same set of columns. Oracle is a relational database management system. Other types of database systems are called hierarchical or network database systems.

remark

In SQL*Plus, a comment you can insert into a command file with the REMARK command.

remote computer

A computer on a network other than the local computer.

remote database

A database other than your default database, which may reside on a remote computer; in particular, one that you reference in the CONNECT, COPY, and SQLPLUS commands.

report

(1) The results of a query. (2) Any output, but especially output that has been formatted for quick reading, in particular, output from SQL*Plus.

reserved word

(1) A word that has a special meaning in a particular software or operating system. (2) In SQL, a set of words reserved for use in SQL statements; you cannot use a reserved word as the name of a database object.

roles

Named groups of related privileges that are granted to users or other roles.

rollback

To discard pending changes made to the data in the current transaction using the SQL ROLLBACK command. You can roll back a portion of a transaction by identifying a savepoint.

row

(1) Synonym for record; one row of data in a database table, having values for one or more columns. Also called tuple. (2) One set of field values in the output of a query. See also column.

schema

A collection of logical structures of data, or schema objects. A schema is owned by a database user and has the same name as that user.

security level

The combination of a hierarchical classification and a set of non-hierarchical compartments that represent the sensitivity of information.

select

To fetch rows from one or more database tables using a query (the SQL statement SELECT).

SELECT list

The list of items that follow the keyword SELECT in a query. These items may include column names, SQL functions, constants, pseudo-columns, calculations on columns, and aliases. The number of columns in the result of the query will match the number of items in the SELECT list.

SELECT statement

A SQL statement that specifies which rows and columns to fetch from one or more tables or views. See also SQL statement.

Server

Oracle software that handles the functions required for concurrent, shared data access to an Oracle database. The server portion receives and processes SQL and PL/SQL statements originating from client applications. The computer that manages the server portion must be optimized for its duties.

session

The time after a username connects to an Oracle database and before disconnecting, and the events that happen in that time.

SET command variable

See system variable.

SGA

See also System Global Area (SGA).

spooling

Sending or saving output to a disk storage area. Often used in order to print or transfer files. The SQL*Plus SPOOL command controls spooling.

SQL (Structured Query Language)

The internationally accepted standard for relational systems, covering not only query but also data definition, manipulation, security and some aspects of referential integrity. See also Data Manipulation Language (DML), Data Definition Language (DDL), and Data Control Language (DCL).

SQL buffer

The default buffer containing your most recently entered SQL command or PL/SQL block. SQL*Plus commands are not stored in the SQL buffer.

SQL command

See SQL statement.

SQL script

A file containing SQL statements that you can run in SQL*Plus to perform database administration quickly and easily.

SQL statement

A complete command or statement written in the SQL language. Synonymous with statement (SQL).

SQL*Loader

An Oracle tool used to load data from operating system files into Oracle database tables.

SQL*Net

Net8's precursor. An Oracle product that works with the Oracle Server and enables two or more computers that run the Oracle RDBMS or Oracle tools such as SQL*Forms to exchange data through a network. SQL*Net supports distributed processing and distributed database capability. SQL*Net runs over and interconnects many communications protocols.

SQL*Plus

An interactive SQL-based language for data manipulation, data definition and the definition of access rights for an Oracle database. Often used as an end-user reporting tool.

statement (SQL)

A SQL statement, and analogous to a complete sentence, as opposed to a phrase. Portions of SQL statements or commands are called expressions, predicates, or clauses. See also SQL statement.

string

Any sequence of words or characters on a line.

substitution variable

In SQL*Plus, a variable name or numeral preceded by one or two ampersands (&). Substitution variables are used in a command file to represent values to be provided when the command file is run.

subtotal

In a report, a total of values in a number column, taken over a group of rows that have the same value in a break field. See also summary.

summary

Summaries, or summary columns, are used to compute subtotals, grand totals, running totals, and other summarizations of the data in a report.

summary line

A line in a report containing totals, averages, maximums, or other computed values. You create summary lines through the BREAK and COMPUTE commands.

syntax

The orderly system by which commands, qualifiers, and parameters are combined to form valid command strings.

SYS username

See also SYSTEM username.

SYSDBA

Privilege that contains all system privileges with the ADMIN OPTION and the SYSOPER system privilege. See also SYSOPER.

SYSOPER

Privilege that allows a DBA to perform operations such as STARTUP, SHUTDOWN, ARCHIVE LOG and RECOVER. See also SYSDBA.

system administrator

A person responsible for operation and maintenance of the operating system of a computer.

system editor

The text editor provided by the operating system.

System Global Area (SGA)

A shared storage area that contains information required by user processes and background processes, such as data and control information for one Oracle instance.

The SGA is allocated when an Oracle instance is started, and is deallocated when the instance shuts down.

SYSTEM username

One of two standard DBA usernames automatically created with each database (the other is SYS). The Oracle user SYSTEM is created with the password MANAGER. The SYSTEM username is the preferred username for DBAs to use when performing database maintenance.

system variable

A variable that indicates status or environment, which is given a default value by Oracle or SQL*Plus. Examples are LINESIZE and PAGESIZE. Use the SQL*Plus commands SHOW and SET to see and alter the value of system variables.

table

The basic unit of storage in a relational database management system. A table represents entities and relationships, and consists of one or more units of information (rows), each of which contains the same kinds of values (columns). Each column is given a column name, a datatype (such as CHAR, NCHAR, VARCHAR2, NVARCHAR2, DATE, or NUMBER), and a width (the width may be predetermined by the datatype, as in DATE). Once a table is created, valid rows of data can be inserted into it. Table information can then be queried, deleted, or updated. To enforce defined business rules on a table's data, integrity constraints and triggers can also be defined for a table.

table alias

A temporary substitute name for a table, defined in a query and only good during that query. If used, an alias is set in the FROM clause of a SELECT statement and may appear in the SELECT list. See also alias.

text editor

A program run under your host computer's operating system that you use to create and edit host system files and SQL*Plus command files containing SQL commands, SQL*Plus commands, and/or PL/SQL blocks.

timer

An internal storage area created by the TIMING command.

title

One or more lines that appears at the top or bottom of each report page. You establish and format titles through the TTITLE and BTITLE commands.

transaction

A logical unit of work that comprises one or more SQL statements executed by a single user. According to the ANSI/ISO SQL standard, with which Oracle is compatible, a transaction begins with the user's first executable SQL statement. A transaction ends when it is explicitly committed or rolled back by the user.

truncate

To discard or lose one or more characters from the beginning or end of a value, whether intentionally or unintentionally.

type

A column contains information in one of four types: character, date, number or long. The operations users can perform on the contents of a column depend on the type of information it contains. See also format.

USERID

A command line argument that allows you to specify your Oracle username and password with an optional Oracle Net address.

username

The name by which a user is known to the Oracle database server and to other users. Every username is associated with a private password, and both must be entered to connect to an Oracle database. See also account.

user variable

A variable defined and set by you explicitly with the DEFINE command or implicitly with an argument to the START command.

VARCHAR

An Oracle Corporation datatype. Specifically, this datatype functions identically to the Oracle VARCHAR2 datatype (see definition below). However, Oracle Corporation recommends that you use VARCHAR2 instead of VARCHAR because Oracle Corporation may change the functionality of VARCHAR in the future.

VARCHAR2

An Oracle Corporation datatype. Specifically, it is a variable-length, alpha-numeric string with a maximum length of 4000 bytes. If data entered for a column of type VARCHAR2 is less than 4000 bytes, no spaces will be padded; the data is stored with a length as entered. If data entered is more than 4000 bytes, an error occurs. For more information about the VARCHAR2 datatype, refer to the *Oracle9i SQL Reference*.

variable

A named object that holds a single value. SQL*Plus uses bind substitution, system, and user variables.

view

A view can be thought of as a "stored query" presenting data from one or many tables. A view does not actually contain or store data, but derives data from the base tables on which it is based. Views can be queried, updated, inserted into, and deleted from. Operations on a view affect the view's base tables.

width

The width of a column, parameter, or layout object. Width is measured in characters; a space is a character.

wrapping

A reporting or output feature in which a portion of text is moved to a new line when the entire text does not fit on one line.

Index

/ (slash) command
 default logon, 5-40
 entered at buffer line-number prompt, 5-9
 entered at command prompt, 5-9
 similar to RUN, 5-9, 5-75
 usage, 5-9

pound sign, 5-28

* (asterisk)
 in LIST command, 5-57

@ ("at" sign)
 command, 5-5
 command file, 5-5
 in CONNECT command, 5-40
 in COPY command, 5-42
 passing parameters to a command file, 5-5
 similar to START, 5-6, 5-104

@@ (double "at" sign) command, 5-7
 command file, 5-7
 similar to START, 5-7, 5-104

~ infinity sign, 5-28

-- negative infinity sign, 5-28

A

ABORT mode, 5-101

ACCEPT command
 and DEFINE command, 5-46

access, denying and granting, A-2

ALIAS clause, 5-24
 in ATTRIBUTE command, 5-13

ALL clause, 5-96

ALTER command
 disabling, A-5

ampersands (&)
 in parameters, 5-5, 5-103

ANALYZE command
 disabling, A-5

APPEND clause
 in COPY command, 5-43

APPINFO clause, 5-77

ARCH background process, 5-11

ARCHIVE LOG
 command, 5-10

argument
 in START command, 5-103

ARRAYSIZE variable, 5-78
 relationship to COPY command, 5-44

ATTRIBUTE command, 5-13
 ALIAS clause, 5-13
 and CLEAR COLUMN command, 5-14
 CLEAR clause, 5-13
 clearing columns, 5-21, 5-24
 controlling display characteristics, 5-14
 display characteristics, 5-13
 entering multiple, 5-14
 FORMAT clause, 5-14
 LIKE clause, 5-14
 listing all attributes' display characteristics, 5-13
 listing an attribute's display characteristics, 5-13
 OFF clause, 5-14
 ON clause, 5-14
 restoring column display attributes, 5-14
 suppressing column display attributes, 5-14

AUDIT command
 disabling, A-5

AUTOCOMMIT variable, 5-78

AUTOMATIC clause, 5-63

AUTOPRINT variable, 5-79
AUTORECOVERY variable, 5-79
AUTOTRACE variable, 5-79

B

background process
 startup after abnormal termination, 5-101
BEGIN command
 disabling, A-5
bind variables
 creating, 5-115
 displaying, 5-59
 displaying automatically, 5-79, 5-117
 in PL/SQL blocks, 5-117
 in SQL statements, 5-117
 in the COPY command, 5-117
blocks, PL/SQL
 setting character used to end, 5-80
 timing statistics, 5-87
BLOCKTERMINATOR, 5-80, 5-87
BOLD clause, 5-73, 5-111
break columns, 5-15
BREAK command, 5-15
 and SQL ORDER BY clause, 5-16
 DUPLICATES clause, 5-18
 listing current break definition, 5-18
 ON column clause, 5-15
 ON expr clause, 5-17
 ON REPORT clause, 5-17
 ON ROW clause, 5-17
 removing definition, 5-21
 SKIP clause, 5-18
 SKIP PAGE clause, 5-18
 specifying multiple break columns, 5-16
 used to format a REFCURSOR variable, 5-117
 used with COMPUTE, 5-15, 5-17, 5-36
break definition
 listing current, 5-18
 removing current, 5-21
BREAKS clause, 5-21
browse button, 3-4
BTITLE clause, 5-97
BTITLE command, 5-20
 aligning title elements, 5-111

BOLD clause, 5-111
CENTER clause, 5-111
COL clause, 5-111
FORMAT clause, 5-111
 indenting titles, 5-111
LEFT clause, 5-111
OFF clause, 5-111
 referencing column value variable, 5-29
RIGHT clause, 5-111
SKIP clause, 5-111
 suppressing current definition, 5-111
TAB clause, 5-111
TTITLE command, 5-20
buffer
 deleting all lines, 5-21
 executing contents, 5-9, 5-75
 listing a range of lines, 5-57
 listing a single line, 5-57
 listing all lines, 5-57
 listing contents, 5-57
 listing the current line, 5-57
 listing the last line, 5-57
 See SQL buffer
BUFFER clause, 5-21
buttons
 browse, 3-4
 cancel, 3-6, 4-5
 change password, 3-6
 clear, 3-3, 3-6
 clear screen, 3-5
 execute, 3-5
 load script, 3-5
 log in, 3-3
 save script, 3-5
 submit for execution, 4-5

C

C:\ORANT, defined, Preface-xvi
C:\ORAWIN95, defined, Preface-xvi
cancel button, 3-6, 4-5
CANCEL clause, 5-64, 5-66
cascading style sheet
 customizing, 2-5
 user interface, 2-5

- CENTER clause, 5-73, 5-111
- change password button, 3-6
- CHAR clause
 - VARIABLE command, 5-115
- CHAR columns
 - changing format, 5-25
 - default format, 5-25
- character encoding
 - national language support, 2-13
- clear button, 3-3, 3-6
- CLEAR clause, 5-24
 - in ATTRIBUTE command, 5-13
- CLEAR command, 5-21
 - BREAKS clause, 5-21
 - BUFFER clause, 5-21
 - COLUMNS clause, 5-21
 - COMPUTES clause, 5-21
 - SQL clause, 5-21
 - TIMING clause, 5-21
- clear screen button, 3-5
- client tier, 1-2
 - configuring, 2-3
- CLOB clause
 - VARIABLE command, 5-116
- CLOB columns
 - changing format, 5-25
 - default format, 5-25
 - setting maximum width, 5-84
 - setting retrieval position, 5-84
 - setting retrieval size, 5-84
- CMDSEP variable, 5-80
- COL clause, 5-73, 5-111
- COLUMN command, 5-23
 - ALIAS clause, 5-24
 - and BREAK command, 5-17
 - and DEFINE command, 5-46
 - CLEAR clause, 5-24
 - displaying column values in bottom titles, 5-29
 - displaying column values in top titles, 5-28
 - entering multiple, 5-30
 - ENTMAP clause, 5-24
 - FOLD_AFTER clause, 5-24, 5-25
 - FOLD_BEFORE clause, 5-25
 - FORMAT clause, 5-25
 - formatting NUMBER columns, 5-26
 - HEADING clause, 5-28
 - HEADSEP character, 5-28
 - JUSTIFY clause, 5-28
 - LIKE clause, 5-28
 - listing column display attributes, 5-23
 - NEW_VALUE clause, 5-28
 - NEWLINE clause, 5-28
 - NOPRINT clause, 5-29
 - NULL clause, 5-29
 - OFF clause, 5-30
 - OLD_VALUE clause, 5-29
 - ON clause, 5-30
 - PRINT clause, 5-29
 - resetting to default display, 5-24
 - restoring column display attributes, 5-30
 - storing current date in variable for titles, 5-31
 - suppressing column display attributes, 5-30
 - TRUNCATED clause, 5-30
 - used to format a REFCURSOR variable, 5-117
 - WORD_WRAPPED clause, 5-30
 - WRAPPED clause, 5-30
- column headings
 - aligning, 5-28
 - changing, 5-28
 - changing character used to underline, 5-87
 - changing to two or more words, 5-28
 - displaying on more than one line, 5-28
 - suppressing printing in a report, 5-82
 - when truncated, 5-25
- columns
 - assigning aliases, 5-24
 - computing summary lines, 5-34
 - copying display attributes, 5-14, 5-28
 - copying values between tables, 5-42
 - displaying values in bottom titles, 5-29
 - displaying values in top titles, 5-28
 - formatting CHAR, VARCHAR, LONG, and DATE, 5-25
 - formatting in reports, 5-23
 - formatting MSLABEL, RAW MSLABEL, ROWLABEL, 5-25
 - formatting NUMBER, 5-26
 - listing display attributes for all, 5-23
 - listing display attributes for one, 5-23
 - names in destination table when copying, 5-43

columns (continued)

- printing line after values that overflow, 5-85
- resetting a column to default display, 5-24
- resetting all columns to default display, 5-21
- restoring display attributes, 5-14, 5-30
- setting printing to off or on, 5-29
- starting new lines, 5-28
- storing values in variables, 5-28
- suppressing display attributes, 5-14, 5-30
- truncating display for all when value overflows, 5-87
- truncating display for one when value overflows, 5-30
- wrapping display for all when value overflows, 5-87
- wrapping display for one when value overflows, 5-30

COLUMNS clause, 5-21

command files

- in @ ("at" sign) command, 5-5
- in @@ (double "at" sign) command, 5-7
- in START command, 5-103
- including comments in, 5-69
- passing parameters to, 5-5, 5-103
- registering, 5-77
- running, 5-5, 5-103
- running nested, 5-7
- uniform resource identifier, 5-5, 5-7, 5-103

commands

- collecting timing statistics on, 5-108
- disabling, A-4
- entering in iSQL*Plus, 3-4, 4-2
- listing current in buffer, 5-59
- re-enabling, A-4
- SQL
 - executing current, 5-9, 5-75
 - setting character used to end and run, 5-87
- SQL*Plus
 - command summary, 5-2

comments

- including in command files, 5-69
- using REMARK to create, 5-69

COMMIT clause

- WHENEVER OSERROR, 5-122
- WHENEVER SQLERROR, 5-123

communication

- between tiers, 1-3
- COMPATIBILITY variable, 5-80
- COMPUTE command, 5-34
 - LABEL clause, 5-35
 - listing all definitions, 5-36
 - maximum LABEL length, 5-35
 - ON column clause, 5-35
 - ON expr clause, 5-35
 - ON REPORT clause, 5-35
 - ON ROW clause, 5-35
 - referencing a SELECT expression in OF, 5-35
 - referencing a SELECT expression in ON, 5-36
 - removing definitions, 5-21
 - used to format a REFCURSOR variable, 5-117

COMPUTES clause, 5-21

CONCAT variable, 5-80

configuration

- client tier, 2-3
- database tier, 2-8
- file associations, 2-3
- globalization support
- iSQL*Plus Server, 2-4
- middle tier, 2-4
- MIME types, 2-3
- Oracle HTTP Server, 2-4
- Oracle9i, 2-8
- testing Oracle HTTP Server configuration file, 2-7

CONNECT command, 5-40

- and @ ("at" sign), 5-40
- changing password, 5-41
- SYSDBA clause, 5-40
- SYSOPER clause, 5-40
- username/password, 5-40

CONNECT command (SQL)

- disabling, A-5
- connect identifier, 2-5, 3-2, 5-40
- field, 3-2
 - in CONNECT command, 5-40
 - in COPY command, 5-42
 - in DESCRIBE command, 5-48

CONTINUE clause

- WHENEVER OSERROR, 5-122
- WHENEVER SQLERROR, 5-123

- COPY command, 5-42
 - and @ ("at" sign), 5-42
 - and ARRAYSIZE variable, 5-44
 - and COPYCOMMIT variable, 5-44
 - and LONG variable, 5-44
 - APPEND clause, 5-43
 - CREATE clause, 5-43
 - creating a table, 5-43
 - destination table, 5-43
 - determining source rows and columns, 5-44
 - disabling, A-5
 - FROM clause, 5-42
 - INSERT clause, 5-43
 - inserting data in a table, 5-43
 - mandatory connect identifier, 5-43
 - naming the source table with SELECT, 5-44
 - query, 5-44
 - REPLACE clause, 5-43
 - replacing data in a table, 5-43
 - specifying columns for destination, 5-43
 - specifying the data to copy, 5-44
 - TO clause, 5-42
 - username/password, 5-42
 - USING clause, 5-44
 - when a commit is performed, 5-44
- COPYCOMMIT variable, 5-81
 - relationship to COPY command, 5-44
- COPYTYPECHECK variable, 5-81
- CREATE clause
 - in COPY command, 5-43
- CREATE command
 - disabling, A-5
- creating the PRODUCT_USER_PROFILE table, A-2
- CSS
 - user interface, 2-5
- cursor variables, 5-117

D

- database
 - connect identifier, 5-40
 - mounting, 5-105
 - opening, 5-106
- database changes, saving automatically, 5-78

- DATABASE clause, 5-64
- database files
 - recovering, 5-62
- database name at startup, 5-105
- database tier, 1-2, 1-3
 - configuration, 2-8
- databases
 - connecting to default, 5-40
 - connecting to remote, 5-40
 - copying data between, 5-42
 - disconnecting without leaving SQL*Plus, 5-54
 - recovering, 5-62
- DATAFILE clause, 5-65
- DATE columns
 - changing format, 5-26, 5-32
- date, storing current in variable for titles, 5-28, 5-31
- DB2, 5-81
- DBA mode, 5-105
- DBA privilege, 5-105
- DBMS_APPLICATION_INFO package, 5-77
- DECLARE command
 - disabling, A-5
- DEFINE command, 5-46
 - and host system editor, 5-47
 - and UNDEFINE command, 5-114
 - CHAR values, 5-46
 - SET DEFINE ON|OFF, 5-81
 - substitution variables, 5-46
- define substitution variables field, 4-5
- Define Substitution Variables screen, 4-5
- DEFINE variable, 5-81
- DELETE command
 - disabling, A-5
- DESCRIBE command (SQL*Plus), 5-48
 - connect_identifier, 5-48
 - PL/SQL properties listed by, 5-49
 - table properties listed by, 5-48
- disabling iSQL*Plus, 2-6
- DISABLED keyword, disabling commands, A-4
- disabling
 - PL/SQL commands, A-5
 - SQL commands, A-4
 - SQL*Plus commands, A-4
- DISCONNECT command, 5-54

- DROP command
 - disabling, A-5
- DUPLICATES clause, 5-18
- dynamic reports
 - creating, 4-6

E

- ECHO variable, 5-81
- editing
 - external editor, 4-2
 - input area, 4-2
 - scripts, 4-2
- editor, 4-2
- EMBEDDED variable, 5-82
- enabling iSQL*Plus, 2-6
- Enter statements field, 3-4
- ENTMAP clause, 5-24
- escape characters, definition of, 5-82
- ESCAPE variable, 5-82
- execute button, 3-5
- EXECUTE command, 5-55
 - disabling, A-5
- executing
 - last statement, 4-3
 - statements, 4-3
- executing scripts
 - See running
- execution statistics
 - including in report, 5-79
- exiting
 - iSQL*Plus, 4-9

F

- FEEDBACK variable, 5-82
- fields
 - connection identifier, 3-2
 - define substitution variables, 4-5
 - enter statements, 3-4
 - new password, 3-6
 - old password, 3-6
 - output options list, 3-5
 - password, 3-2
 - privilege, 3-3

- retype new password, 3-6
- script location, 3-4
- username, 3-2, 3-6
- file
 - configuring associations, 2-3
 - output, 3-5
- file names
 - in @ ("at" sign) command, 5-5
 - in @@ (double "at" sign) command, 5-7
 - in START command, 5-103
- FLAGGER variable, 5-82
- FOLD_AFTER clause, 5-25
- FOLD_BEFORE clause, 5-25
- font
 - national language support, 2-12
- footers
 - aligning elements, 5-73
 - displaying at bottom of page, 5-70
 - displaying system-maintained values, 5-72
 - formatting elements, 5-73
 - indenting, 5-73
 - listing current definition, 5-70
 - suppressing definition, 5-73
- FORCE clause, 5-105
- FORMAT clause, 5-25
 - in ATTRIBUTE command, 5-14
 - in REPHEADER and REPFOOTER commands, 5-73
 - in TTITLE and BTITLE commands, 5-111
- format models, number, 5-27
- FROM clause, 5-63
- FROM clause (SQL*Plus), 5-42
- functionality, 1-2

G

- globalization support
 - Oracle9i database, 2-14
- glogin.sql, 2-12
- GRANT command, A-2
 - disabling, A-5

H

headers

- displaying at top of page, 5-72
- displaying system-maintained values, 5-72

HEADING clause, 5-28

HEADING variable, 5-82

headings

- aligning elements, 5-73
- column headings, 5-82
- formatting elements, 5-73
- indenting, 5-73
- listing current definition, 5-74
- suppressing definition, 5-73

HEADSEP variable, 5-83

- use in COLUMN command, 5-28

HELP command, 5-56

help, online, 5-56

HTML

- running dynamic reports, 4-6

HTTP

- HTTPS security, 2-9
- security, 2-8

httpd.conf, 2-7

- testing httpd.conf edits, 2-6

I

IMMEDIATE mode, 5-101

infinity sign (~), 5-28

initialization parameters

- displaying, 5-97

INIT.ORA file

- parameter file, 5-105

input area, 3-4

- editing, 4-2

- entering statements, 4-2

INSERT clause, 5-43

INSERT command

- disabling, A-5

installation

- general, 2-2
- required components, 2-2

INSTANCE variable, 5-83

instances

- shutting down, 5-101

- starting, 5-105

iplus.css, 2-6

iSQL*Plus, 1-2

- cascading style sheet, 2-5

- enabling or disabling, 2-6

- entering commands, 3-4, 4-2

- exiting, 4-9

- log in screen, 4-2

- overview, 1-2

- Server configuration, 2-4

- session, 1-3

- session timeout, 2-5

- starting, 4-2

- three-tier model, 1-2

- user interface, 1-2, 1-3

iSQL*Plus Server

- defining a connection identifier, 2-5

isqlplus URI, 4-2

isqlplus.conf, 2-5, 2-7

isqlplusdba URI, 4-2

J

Japanese, 2-12

JUSTIFY clause, 5-28

L

labels

- in COMPUTE command, 5-35

last statement

- executing, 4-3

LEFT clause, 5-73, 5-111

LIKE clause, 5-14, 5-28

lines

- changing width, 5-83

- listing all in buffer, 5-57

LINESIZE variable, 5-83

LIST clause, 5-10

LIST command, 5-57

- determining current line, 5-57

- making last line current, 5-57

- using an asterisk, 5-57

- LNO clause, 5-97
- load script button, 3-5
- LOBOFFSET variable, 5-84
- LOCK TABLE command
 - disabling, A-5
- log in button, 3-3
- Log In screen, 3-2, 4-2
- Log Out, 4-9
- LOG_ARCHIVE_DEST parameter, 5-10
- LOGFILE clause, 5-63
- logging off
 - conditionally, 5-122, 5-123
 - Oracle, 5-54
- logging on
 - Oracle, 5-40
- login.sql
 - See glogin.sql
- LONG columns
 - changing format, 5-25
 - default format, 5-25
 - setting maximum width, 5-84
 - setting retrieval size, 5-84
- LONG variable, 5-84
 - effect on COPY command, 5-44
- LONGCHUNKSIZE variable, 5-25, 5-84

M

- MARKUP, 5-84
- media recovery, 5-106
- message, sending to screen, 5-61
- Microsoft Internet Explorer
 - configuring proxy exceptions, 2-4
- middle tier, 1-2
 - configuration, 2-4
- MIME type
 - configuration, 2-3
- mod_ssl, 2-9
- MOUNT clause, 5-105
- mounting a database, 5-105

N

- national language support, 2-12
 - character encoding, 2-13

- font, 2-12
 - See also globalization support
- NCHAR clause
 - VARIABLE command, 5-115
- NCHAR columns
 - changing format, 5-25
 - default format, 5-25
- NCLOB clause
 - VARIABLE command, 5-116
- NCLOB columns
 - changing format, 5-25
 - default format, 5-25
 - setting maximum width, 5-84
 - setting retrieval position, 5-84
 - setting retrieval size, 5-84
- negative infinity sign (--), 5-28
- Netscape Navigator
 - configuring proxy exceptions, 2-4
- new password field, 3-6
- NEW_VALUE clause, 5-28
 - storing current date in variable for titles, 5-28
- NEWLINE clause, 5-28
- NEXT clause, 5-11
- NLS, 2-12
- NLS_DATE_FORMAT, 5-32
- NOAUDIT command
 - disabling, A-5
- NOMOUNT clause, 5-106
- NONE clause
 - WHENEVER OSERROR, 5-122
 - WHENEVER SQLERROR, 5-123
- NOPRINT clause, 5-29
- NORMAL mode, 5-101
- NULL clause, 5-29
- null values
 - setting text displayed, 5-29, 5-84
- NULL variable, 5-84
- NUMBER clause
 - VARIABLE command, 5-115
- NUMBER columns
 - changing format, 5-26
 - default format, 5-27
- number formats
 - setting default, 5-85
- NUMFORMAT variable, 5-85

NUMWIDTH variable, 5-85
effect on NUMBER column format, 5-27
NVARCHAR2 columns
changing format, 5-25
default format, 5-25

O

objects, describing, 5-81
OFF clause, 5-30
in ATTRIBUTE command, 5-14
in COLUMN command, 5-30
in REPFOOTER commands, 5-73
in REPHEADER commands, 5-73
in TTITLE and BTITLE commands, 5-111
old password field, 3-6
OLD_VALUE clause, 5-29
ON clause
in ATTRIBUTE command, 5-14
in COLUMN command, 5-30
ON column clause
in BREAK command, 5-15
in COMPUTE command, 5-35
ON expr clause
in BREAK command, 5-17
in COMPUTE command, 5-35
ON REPORT clause
in BREAK command, 5-17
in COMPUTE command, 5-35
ON ROW clause
in BREAK command, 5-17
in COMPUTE command, 5-35
online help, 5-56
OPEN clause, 5-106
opening a database, 5-106
Oracle HTTP Server, 1-2, 1-3, 2-2
configuration, 2-4
glogin.sql, 2-12
Product User Profile table, 2-12
starting and stopping, 2-7
supported, 2-2
testing configuration file, 2-7
Oracle Net, 1-3, 2-9
connect identifier, 5-40
security, 2-9

services, 2-2
oracle_apache.conf, 2-6, 2-7
ORACLE_BASE
explained, Preface-xvi
ORACLE_HOME
explained, Preface-xvi
Oracle9i, 1-3, 2-8
Oracle9i database
globalization support, 2-14
output options
file, 3-5
new window, 3-5
work screen, 3-5
output options list, 3-5

P

PAGE clause, 5-72
pages
changing length, 5-85
PAGESIZE variable, 5-85
parameter files (INIT.ORA files)
specifying alternate, 5-105
parameters, 5-5, 5-103
PARAMETERS clause, 5-97
parsing
httpd.conf file, 2-6
password
in CONNECT command, 5-40
password field, 3-2
Period (.)
terminating PL/SQL blocks, 5-80
PL/SQL
executing, 5-55
formatting output in SQL*Plus, 5-117
PNO clause, 5-98
pound sign (#), 5-28
prerequisites, 2-2
PRINT clause, 5-29
PRINT command, 5-59
printing
bind variables automatically, 5-79
REFCURSOR variables, 5-117

- privileges
 - list, 3-3
 - User, 2-7
- Product User Profile table, 2-12
- PRODUCT_USER_PROFILE table, A-2
- PROMPT command, 5-61
- proxy server exceptions
 - configuration, 2-3
 - Microsoft Internet Explorer, 2-4
 - Netscape Navigator, 2-4
- PUPBLD.SQL, A-2

Q

- queries
 - in COPY command, 5-44
 - show number of records retrieved, 5-82
- query execution path
 - including in report, 5-79

R

- record separators, printing, 5-85
- RECOVER
 - clause, 5-106
- RECOVER command, 5-62
 - AUTOMATIC clause, 5-63
 - CANCEL clause, 5-64, 5-66
 - CONTINUE clause, 5-63
 - DATABASE clause, 5-64
 - FROM clause, 5-63
 - LOGFILE clause, 5-63
 - STANDBY DATABASE clause, 5-64
 - STANDBY DATAFILE clause, 5-65
 - STANDBY TABLESPACE clause, 5-65
 - UNTIL CANCEL clause, 5-64
 - UNTIL CONTROLFILE clause, 5-65
 - UNTIL TIME clause, 5-64
 - USING BACKUP CONTROL FILE clause, 5-64
- recovery
 - RECOVER command, 5-62
- RECSEP variable, 5-85
- RECSEPCCHAR variable, 5-85
- redo Log Files
 - ARCHIVE LOG command, 5-10

- REFCURSOR clause
 - VARIABLE command, 5-117
- RELEASE clause, 5-98
- REMARK command, 5-69
- remote database
 - connect identifier, 3-2
- RENAME command
 - disabling, A-5
- REPFOOTER clause, 5-98
- REPFOOTER command, 5-70
 - aligning footer elements, 5-73
 - BOLD clause, 5-73
 - CENTER clause, 5-73
 - COL clause, 5-73
 - FORMAT clause, 5-73
 - indenting report footers, 5-73
 - LEFT clause, 5-73
 - OFF clause, 5-73
 - RIGHT clause, 5-73
 - SKIP clause, 5-73
 - suppressing current definition, 5-73
 - TAB clause, 5-73
- REPHEADER clause, 5-98
- REPHEADER command, 5-72
 - aligning heading elements, 5-73
 - BOLD clause, 5-73
 - CENTER clause, 5-73
 - COL clause, 5-73
 - FORMAT clause, 5-73
 - indenting headings, 5-73
 - LEFT clause, 5-73
 - OFF clause, 5-73
 - PAGE clause, 5-72
 - RIGHT clause, 5-73
 - SKIP clause, 5-73
 - suppressing current definition, 5-73
 - TAB clause, 5-73
- REPLACE clause
 - in COPY command, 5-43
- report
 - breaks, 5-15
- reports
 - columns, 5-24
 - creating bottom titles, 5-20
 - creating dynamic, 4-6

- reports (continued)
 - creating footers, 5-70
 - creating headers, 5-72
 - creating master/detail, 5-29
 - creating top titles, 5-110
 - displaying, 5-79
 - formatting column headings, 5-23
 - formatting columns, 5-23
 - running from a URI, 4-6
 - starting on a new page, 5-82
 - title, 5-110
 - requirements, 2-2
 - RESTRICT, 5-105
 - retype new password field, 3-6
 - REVOKE command, A-2
 - disabling, A-5
 - RIGHT clause, 5-73, 5-111
 - roles, A-6
 - disabling, A-7
 - re-enabling, A-8
 - ROLLBACK clause
 - WHENEVER OSERROR, 5-122
 - WHENEVER SQLERROR, 5-123
 - rows
 - performing computations on, 5-34
 - setting number retrieved at one time, 5-78
 - setting the number after which COPY commits, 5-81
 - RUN command, 5-75
 - disabling, A-5
 - similar to / (slash) command, 5-75
- S**
-
- save script button, 3-5
 - screens
 - connection identifier field, 3-2
 - define substitution variables, 4-5
 - define substitution variables field, 4-5
 - enter statements field, 3-4
 - Log In, 3-2
 - new password field, 3-6
 - old password field, 3-6
 - output options, 3-5
 - password field, 3-2
 - privilege list, 3-3
 - retype new password field, 3-6
 - script location field, 3-4
 - username field, 3-2, 3-6
 - Work, 3-4
 - script
 - editing, 4-2
 - script location field, 3-4
 - Secure Sockets Layer
 - security, 2-9
 - security
 - HTTP, 2-8
 - Oracle HTTP Server, 2-9
 - Oracle Net, 2-9
 - PRODUCT_USER_PROFILE table, A-2
 - Secure Sockets Layer, 2-9
 - SSL, 2-9
 - SELECT command
 - and BREAK command, 5-16, 5-17
 - and COLUMN command, 5-24
 - and COPY command, 5-44
 - and DEFINE command, 5-46
 - disabling, A-5
 - SERVEROUTPUT variable, 5-86
 - session
 - concurrent, 1-3
 - identification, 1-3, 2-8, 4-2
 - stateful behavior, 2-8
 - SET command, 5-76
 - affecting substitution variables, 4-4
 - APPINFO variable, 5-77
 - ARRAYSIZE variable, 5-78
 - AUTOCOMMIT variable, 5-78
 - AUTOPRINT variable, 5-79, 5-117
 - AUTORECOVERY variable, 5-79
 - AUTOTRACE variable, 5-79
 - BLOCKTERMINATOR variable, 5-80
 - CMDSEP variable, 5-80
 - COMPATIBILITY variable, 5-80
 - CONCAT variable, 5-80
 - COPYCOMMIT variable, 5-81
 - COPYTYPECHECK variable, 5-81
 - DEFINE variable, 5-81
 - DESCRIBE variable, 5-81
 - disabling, A-5

- SET command (continued)
 - ECHO variable, 5-81
 - EMBEDDED variable, 5-82
 - ESCAPE variable, 5-82
 - FEEDBACK variable, 5-82
 - FLAGGER variable, 5-82
 - HEADING variable, 5-82
 - HEADSEP variable, 5-83
 - INSTANCE variable, 5-83
 - LINESIZE variable, 5-83
 - LOBOFFSET variable, 5-84
 - LOGSOURCE variable, 5-84
 - LONG variable, 5-84
 - LONGCHUNKSIZE variable, 5-84
 - MARKUP clause, 5-84
 - NULL variable, 5-84
 - NUMFORMAT variable, 5-85
 - NUMWIDTH variable, 5-27, 5-85
 - PAGESIZE variable, 5-85
 - RECSEP variable, 5-85
 - RECSEPCHAR variable, 5-85
 - SERVEROUTPUT variable, 5-86
 - SQLCASE variable, 5-86
 - SQLPLUSCOMPATIBILITY variable, 5-86
 - SQLTERMINATOR variable, 5-87
 - TIMING variable, 5-87
 - UNDERLINE variable, 5-87
 - used to format a REFCURSOR variable, 5-117
 - VERIFY variable, 5-87
 - WRAP variable, 5-87
- SET ROLE command
 - disabling, A-5
- SET TRANSACTION command
 - disabling, A-5
- SGA clause, 5-98
- SHOW clause, 5-108
- SHOW command, 5-96
 - ALL clause, 5-96
 - BTITLE clause, 5-97
 - ERRORS clause, 5-97
 - LNO clause, 5-97
 - PNO clause, 5-98
 - RELEASE clause, 5-98
 - REPFOOTER clause, 5-98
 - REPHEADER clause, 5-98
 - SPOOL clause, 5-98
 - SQLCODE clause, 5-98
 - TTITLE clause, 5-98
 - USER clause, 5-98
- SHUTDOWN command, 5-101
 - ABORT, 5-101
 - IMMEDIATE, 5-101
 - NORMAL, 5-101
 - TRANSACTIONAL LOCAL, 5-102
- SKIP clause
 - in BREAK command, 5-18
 - in REPHEADER and REPFOOTER commands, 5-73
 - in TTITLE and BTITLE commands, 5-111
- SKIP PAGE clause, 5-18
- slash (/) command, 5-9
- SPOOL clause, 5-98
- SQL buffer, 4-3
- SQL clause, 5-21
- SQL DML statements
 - reporting on, 5-79
- SQL*Plus, 2-2
 - command summary, 5-2
- SQLCASE variable, 5-86
- SQLCODE clause, 5-98
 - SHOW command, 5-98
- SQLNUMBER variable, 5-86
- SQLTERMINATOR variable, 5-87
- SSL
 - security, 2-9
- STANDBY DATAFILE clause, 5-65
- STANDBY TABLESPACE clause, 5-65
- START clause, 5-11, 5-108
- START command, 5-103
 - arguments, 5-103
 - command file, 5-103
 - disabling, A-5
 - passing parameters to a command file, 5-103
 - similar to @ ("at" sign) command, 5-6, 5-104
 - similar to @@ (double "at" sign) command, 5-7, 5-104
- starting
 - iSQL*Plus, 4-2
 - Oracle HTTP Server, 2-7

- STARTUP command, 5-105
 - FORCE clause, 5-105
 - MOUNT clause, 5-105
 - NOMOUNT clause, 5-106
 - OPEN clause, 5-106
 - PFILE clause, 5-105
 - RECOVER clause, 5-106
 - RESTRICT clause, 5-105
 - specifying a database, 5-105
- statements
 - entering in input area, 4-2
 - executing, 4-3
- STOP clause, 5-11, 5-108
- stopping
 - Oracle HTTP Server, 2-7
- submit for execution button, 4-5
- substitution variables, 4-4
 - concatenation character, 5-80
 - DEFINE command, 5-46
 - prefixing, 5-81
- summary lines
 - computing and printing, 5-34
- supported
 - web browser, 2-2
 - web server, 2-2
- SYSDBA clause, 5-40
- SYSOPER clause, 5-40
- system
 - prerequisites, 2-2
- system variables, 5-88
 - changing current settings, 5-76
 - listing current settings, 5-96
- system-maintained values
 - displaying in headers and footers, 5-72
 - displaying in titles, 5-110

T

- TAB clause, 5-73, 5-111
- tables
 - controlling destination when copying, 5-43
 - copying values between, 5-42
 - listing column definitions, 5-48
- TABLESPACE clause, 5-64
- tablespaces, recovering, 5-62

- text, clearing from buffer, 5-21
- three-tier model, 1-2
- timeout
 - iSQL*Plus sessions, 2-5
- TIMING clause, 5-21
- TIMING command, 5-108
 - deleting all areas created by, 5-21
 - deleting current area, 5-108
 - SHOW clause, 5-108
 - START clause, 5-108
 - STOP clause, 5-108
- TIMING variable, 5-87
- titles
 - aligning elements, 5-111
 - displaying at bottom of page, 5-20
 - displaying at top of page, 5-110
 - displaying column values, 5-28, 5-29
 - displaying current date, 5-28, 5-31
 - displaying page number, 5-112
 - displaying system-maintained values, 5-110
 - formatting elements, 5-111
 - indenting, 5-111
 - listing current definition, 5-20, 5-112
 - setting lines from top title to end of page, 5-85
 - setting top and bottom, 5-20, 5-110
 - suppressing definition, 5-111
- TNS alias, 3-2
- tnsnames.ora, 2-5
- TO clause, 5-42
- TRUNCATE command
 - disabling, A-5
- TRUNCATED clause, 5-30
- TTITLE clause, 5-98
- TTITLE command, 5-110
 - aligning title elements, 5-111
 - BOLD clause, 5-111
 - CENTER clause, 5-111
 - COL clause, 5-111
 - FORMAT clause, 5-111
 - indenting titles, 5-111
 - LEFT clause, 5-111
 - listing current definition, 5-112
 - OFF clause, 5-111
 - referencing column value variable, 5-28
 - RIGHT clause, 5-111

TTITLE command (continued)
SKIP clause, 5-111
suppressing current definition, 5-111
TAB clause, 5-111

U

UNDEFINE command, 5-114
and DEFINE command, 5-46
UNDERLINE variable, 5-87
unicode, 2-12
UNTIL CANCEL clause, 5-64
UNTIL CHANGE clause, 5-64
UNTIL CONTROLFILE clause, 5-65
UNTIL TIME clause, 5-64
UPDATE command
disabling, A-5
URI
isqlplus, 4-2
isqlplusdba, 4-2
running dynamic reports, 4-6
URL, 1-3
USER clause, 5-98
user interface, 1-2
supported, 2-2
User privilege, 2-7
user variables
defining, 5-46
deleting, 5-114
displaying in headers and footers, 5-72
displaying in titles, 5-110
listing definition of one, 5-46
listing definitions of all, 5-46
username
connecting under different, 5-40
field, 3-2, 3-6
in CONNECT command, 5-40
USING BACKUP CONTROL FILE clause, 5-64
USING clause, 5-44
UTF-8, 2-12

V

V\$SESSION virtual table, 5-77
V\$SQLAREA virtual table, 5-77

VARCHAR columns
default format, 5-25
VARCHAR2 clause
VARIABLE command, 5-116
VARCHAR2 columns
changing format, 5-25
VARIABLE command, 5-115
CHAR clause, 5-115
CLOB clause, 5-116
NCHAR clause, 5-115
NCLOB clause, 5-116
NUMBER clause, 5-115
REFCURSOR clause, 5-117
VARCHAR2 clause, 5-116
variable clause, 5-115
variables
substitution, 4-4
user variables, 5-46
VERIFY variable, 5-87

W

web browser, 1-2
supported, 2-2
web server, 1-2
WHENEVER OSERROR command, 5-122
COMMIT clause, 5-122
CONTINUE clause, 5-122
NONE clause, 5-122
ROLLBACK clause, 5-122
WHENEVER SQLERROR command, 5-123
COMMIT clause, 5-123
CONTINUE clause, 5-123
NONE clause, 5-123
ROLLBACK clause, 5-123
window output, 3-5
WORD_WRAPPED clause, 5-30
Work screen, 3-4
output options, 3-5
WRAP variable, 5-87
WRAPPED clause, 5-30