

AT&T 585-350-917
Issue 1
Comcode 407298686

SQL*REPORTWRITER[®]
DOCUMENTATION SET

VERSION 1.1

ORACLE[®]

The Relational Database Management System

INTRODUCTION TO
SQL * REPORTWRITER™
V1.1

ORACLE®

The Relational Database Management System

Introduction to SQL*ReportWriter V1.1

Part No. 19677-0689

Copyright © 1988,1989 Oracle Corporation, Belmont, California

All rights reserved. Printed in the U.S.A.

Restricted Rights Legend

Use, duplication, or disclosure is subject to restrictions stated in your contract with Oracle Corporation.

Use, duplication, or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal Law.

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.

ORACLE is a registered trademark and SQL*ReportWriter a trademark of Oracle Corporation.



PREFACE

SQL*ReportWriter is Oracle Corporation's flagship reporting product. Designed for the application developer who is familiar with SQL, SQL*ReportWriter enables users to develop a wide variety of reports quickly and easily.

Since 1977, Oracle Corporation has served the database needs of businesses and government agencies worldwide. In 1979, Oracle produced its primary product the ORACLE Relational Database Management System. ORACLE was the first product based on IBM's SQL database language, and it remains the most powerful relational database available today.

Oracle Corporation offers an integrated family of CASE Tools that includes tools for reports, forms, menus, graphics, and database design. SQL*ReportWriter is the reporting component of Oracle's CASE Tools.

The purpose of this Introduction is to give you a basic understanding of SQL*ReportWriter's capabilities. You'll also see what it is like to build and maintain reports yourself. If you would like to have further information about SQL*ReportWriter, the *SQL*ReportWriter Reference Manual*, Part No. 641-V1.1, is also available in addition to this Introduction. It contains more details about SQL*ReportWriter concepts, a tutorial to help you get started using SQL*ReportWriter, and a reference section organized around each of the SQL*ReportWriter screens.



CONTENTS

CHAPTER 1	WELCOME TO SQL*REPORTWRITER	1 - 1
	About SQL*ReportWriter	1 - 2
	The Application	1 - 4
CHAPTER 2	DESIGNING A REPORT	2-1
	SQL*ReportWriter Screens	2 - 2
	Report Building Process	2 - 2
	Naming a Report	2 - 3
	Selecting Data	2 - 4
	Executing the Report	2 - 7
	Using the Help System	2 - 8
	Modifying the Default Report	2 - 9
	Chapter Summary	2-19
CHAPTER 3	FLEXIBLE REPORTING	3-1
	The Master/Detail Report	3 - 2
	Production Reporting	3-13
	Chapter Summary	3-17

CHAPTER 4	POWERFUL REPORTING	4-1
	The Matrix Report	4 - 2
	The Form Letter Report	4 - 6
	The Mailing Label Report	4-10
	The Master/Detail/Summary Report	4-12
	Product Integration	4-17
CHAPTER 5	SUMMARY	5-1

1

WELCOME TO SQL*REPORTWRITER

The Introduction to SQL*ReportWriter illustrates how SQL*ReportWriter can help you create any kind of report, quickly and easily. You will see the power of SQL*ReportWriter through the development of a series of reports. We will start with a simple example and build up to more complex reports.

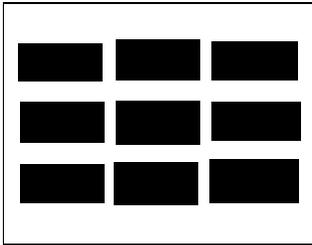
In Chapter 1, we introduce you to SQL*ReportWriter's features and to the application used in this manual.

In Chapter 2, we design a tabular report using only SQL*ReportWriter's default settings. We then modify this report by adding a page heading and a summary. After reading this chapter, you will have a basic understanding of the important SQL*ReportWriter concepts.

In Chapter 3, you learn how to make major changes to a report with just a few keystrokes. We modify the tabular report created in Chapter 2 by creating a new group, adding new data, modifying fields, positioning text, and more.

Finally, in Chapter 4, we demonstrate a few of the additional kinds of reports that you can create using SQL*ReportWriter. We build a matrix report, a form letter report, a mailing label report, and a complex multi-part report.

Mailing Labels



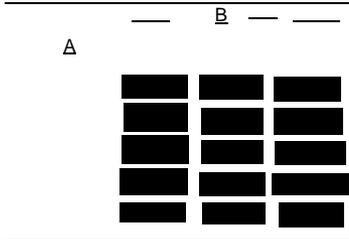
Print labels in multiple columns.

Letter



Mix text and data to create personalized form letters.

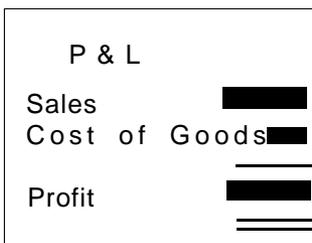
Crosstab, Matrix, Across



	A	B	
A	[redacted]	[redacted]	[redacted]
	[redacted]	[redacted]	[redacted]

Create a matrix report simply by specifying the columns, the rows, and the information that goes in the cells.

Financial



P & L	
Sales	[redacted]
Cost of Goods	[redacted]
Profit	[redacted]

Format data and boilerplate text with absolute precision.

The Application

The data used in the following examples comes from a fictitious sporting goods manufacturer named Summit Sporting Goods. Four tables from Summit's database are used: the ORD, ITEM, PRODUCT, and CUSTOMER tables.

We use the ORD table in the first example report to list information about Summit's orders. In the second example, we add the ITEM table to list data about the items within each order. The PRODUCT and CUSTOMER tables are used in Chapter 4 to create reports about the products and customers at Summit.

The ORD Table

<u>Ordid</u>	<u>Orderdate</u>	<u>Complan</u>	<u>Custid</u>	<u>Shipdate</u>	<u>Total</u>
6 1 0	07 - JAN - 87	A	1 0 1	08-JAN-87	101.40
6 1 1	11 - JAN - 87	B	1 0 2	11-JAN-87	45.00
6 1 2	15 - JAN - 87	C	1 0 4	20-JAN-87	5310.00
6 0 1	01 - MAY - 86	A	1 0 6	30-MAY-86	2.40
6 0 2	05 - JUN - 86	B	1 0 2	20-JUN-86	56.00
6 0 4	15 - JUN - 86	A	1 0 6	30-JUN-86	698.00
6 0 5	14 - JUL - 86	A	1 0 6	30-JUL-86	8674.00
6 0 6	14 - JUL - 86	A	1 0 0	30-JUL-86	3.40
6 0 9	01 - AUG - 86	B	1 0 0	15-AUG-86	97.50
6 0 7	18 - JUL - 86	C	1 0 4	18-JUL-86	5.60
6 0 8	25 - JUL - 86	C	1 0 4	25-JUL-86	35.20
6 0 3	05 - JUN - 86		1 0 2	05-JUN-86	224.00
6 2 0	12 - MAR - 87		1 0 0	12-MAR-87	4450.00
6 1 3	01 - FEB - 87		1 0 8	01-FEB-87	10760.00
6 1 4	01 - FEB - 87		1 0 2	05-FEB-87	23940.00
6 1 6	03 - FEB - 87		1 0 3	10-FEB-87	764.00
6 1 9	22 - FEB - 87		1 0 4	04-FEB-87	980.00
6 1 7	05 - FEB - 87		1 0 5	03-MAR-87	46370.00
6 1 5	01 - FEB - 87		1 0 7	06-FEB-87	710.00
6 1 8	15 - FEB - 87	A	1 0 2	06-MAR-87	3555.50

The column definitions are:

ORDID	order identification number
ORDERDATE	date the order was issued
COMMPLAN	commission plan used
CUSTID	customer identification number
SHIPDATE	date the order was shipped
TOTAL	total price of all the items in the order

The ITEM Table

<u>Ordid</u>	<u>Itemid</u>	<u>Prodid</u>	<u>Actualprice</u>	<u>Qty</u>	<u>Itemtot</u>
6 1 0	3	100890	58	1	58.00
6 1 1	1	100861	45	1	45.00
6 1 2	1	100860	30	100	3000.00
6 0 1	1	200376	2.4	1	2.40
6 0 2	1	100870	2.8	20	56.00
6 0 4	1	100890	58	3	174.00
6 0 4	2	100861	42	2	84.00
6 0 4	3	100860	44	10	440.00
6 0 3	2	100860	56	4	224.00
6 1 0	1	100860	35	1	35.00
6 1 0	2	100870	2.8	3	8.40
6 1 3	4	200376	2.2	200	440.00
6 1 4	1	100860	35	444	15540.00
6 1 4	2	100870	2.8	1000	2800.00
6 1 2	2	101861	40.5	20	810.00
6 1 2	2	100863	10	150	1500.00
6 1 2	3	100863	10	150	1500.00
6 2 0	1	200360	35	10	350.00
6 2 0	1	200376	2.4	1000	2400.00
6 2 0	2	200376	2.4	1000	2400.00
6 2 0	2	102130	3.4	500	1700.00
6 2 0	3	102130	3.4	500	1700.00
6 1 3	1	100871	5.6	100	560.00

- ORDID order identification number for this item
- ITEMID identification number of the line item in the order
- PRODID product identification number
- ACTUALPRICE price per item charged to the customer
- QTY quantity of the line item ordered
- ITEMTOT total cost of the item

The PRODUCT Table

Prodid	Descrip
100860	ACE TENNIS RACKET I
100861	ACE TENNIS RACKET II
100870	ACE TENNIS BALLS-3 PACK
100871	ACE TENNIS BALLS-6 PACK
100890	ACE TENNIS NET
101860	SP TENNIS RACKET
101863	SP JUNIOR RACKET
102130	RH: "GUIDE TO TENNIS"
200376	SB ENERGY BAR-6 PACK
200380	SB VITA SNACK-6 PACK

PRODID product identification number
DESCRIP description of the product

The CUSTOMER Table

<u>Custid</u>	<u>Name</u>	<u>Address</u>	<u>City</u>	<u>State</u>	<u>Zip</u>
100	JACKSPORTS	345 VIEWRIDGE	BELMONT	CA	96711
101	TKB SPORTS SHOP	490 BOLI RD.	REDWOOD CITY	CA	94061
102	VOLLYRITE	9722 HAMILTON	BURLINGAME	CA	95133
103	JUST TENNIS	HILLVIEW MALL	BURLINGAME	CA	97544
104	EVERY MOUNTAIN	574 SURRY RD.	CUPERTINO	CA	93301
105	K+T SPORTS	3476 EL PASEO	SANTA CLARA	CA	91003
106	SHAPE UP	908 SEQUIA	PALO ALTO	CA	94301
107	WOMENS SPORTS	VALCO VILLAGE	SUNNYVALE	CA	93301
108	NORTH WOODS HE	98 LONE PINE	HIBBING	MN	55649

<u>Area</u>	<u>Phone</u>	<u>Repid</u>	<u>Creditlimit</u>	<u>Comments</u>
415	598-6609	7844	5000	Very friendly people to work w
415	368-1223	7521	10000	Rep called 5/8 about change in
415	644-3341	7654	7000	Company doing haevy promotion
415	677-9312	7521	3000	Contact rep about new line of
488	996-2323	7499	10000	Customer with high market shar
488	376-9966	7844	5000	Tends to order large amount o
415	364-9777	7521	6000	Support intensive. Order smal
408	967-4398	7499	10000	First sporting goods store gea
612	566-9123	7844	8000	

CUSTID	customer identification number
NAME	customer's name
ADDRESS	customer's street address
CITY	city in which the customer lives
STATE	state in which the customer is located
ZIP	customer's zip code
AREA	area code in which the customer is located
PHONE	customer's phone number
REPID	identification number of the customer's representative
CREDITLIMIT	customer's credit limit
COMMENTS	remarks about the customer

CHAPTER

2

DESIGNING A REPORT

In this chapter, we develop a simple tabular report to introduce you to SQL*ReportWriter. We will use each major area of SQL*ReportWriter in building this report to give you an understanding of SQL*ReportWriter concepts and capabilities.

You will see that SQL*ReportWriter's default settings for column width, headings, display formats, and more, let you design a report with just a few keystrokes. SQL*ReportWriter's sophisticated fill-in-the-form interface, with instant validation and hints, guides you through every step. And, by directly entering SQL, the language you already know, you have unrestricted access to your data.

You will also get a brief introduction to SQL*ReportWriter's comprehensive online help system, which has an answer to just about every question. At the end of Chapter 2, you will have been introduced to the concepts needed to build virtually any report.

SQL*ReportWriter Screens

Easy-to-Use Interface

When you enter SQL*ReportWriter, a screen appears with the main menu. The main menu has a set of options that take you to other screens in which you create and manipulate your report.

Define Reports in the
SQL*ReportWriter Screens

Main Menu
Title Line

Hint Line
Status Line

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
SQL*ReportWriter								
Perform global operation on reports.								
Report Name:							<INSERT>	

In addition, SQL*ReportWriter screens contain a title line, hint line, and status line to make defining your reports as simple as possible. All screens except this first screen also provide a work area consisting of a form or spreadsheet.

Report Building Process

Building reports in SQL*ReportWriter involves defining seven different types of objects, each accessed from the main menu. Use the Query choice to select, aggregate, and sort data from the database. The Group choice lets you break your report into sections for layout and subtotalling purposes. Select the Field option to adjust attributes, such as the width and display format of each data item in the report. The Summary option lets you define subtotals and other calculations. Finally, use the Text choice to “fine-tune” the position of the text and parameters in your report. The Report choice accesses the report height, width, and other specifications. The Parameter choice allows you to control your report with data selection and other parameters.

Naming a Report

The first step in building a report is to specify the name of the report. To name a new report, choose the **New** option from the **Action** menu. We've named our report Order_Report.

Choose a Report
with the Action Menu

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
New						SQL*ReportWriter		
Open								
Copy								
Rename								
Drop								
Execute								
Generate								
Quit								

Enter a name for the new report

Report Name: <List><Insert>

The Action menu contains a list of report definition operations, such as opening, executing, or deleting a report. This is the only main menu choice that contains a pull-down menu of different operations. All the other main menu choices take you directly to screens to define the data and layout of your report. Use the **Help** option to access SQL*ReportWriter's help system. At all times, only valid choices are accessible.

Selecting Data

After naming a report, select the data you need in the Query Screen. When you choose Query from the main menu, the following screen appears

Specify Report Data
in the Query Screen

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name:				SELECT Statement				Query 1 of 1
^								
v								
Parent-Child Relationships								
Parent Query 1:				Parent Query 2:				
	Child Columns	Parent 1 Columns			Parent 2 Columns			
^								
v								
Enter a name for this query.								
Report Name: Order_Report							<Replace>	

The Query Screen lets you specify data using SQL SELECT statements. First, name the query so you can reference it later, then enter the SELECT statement. The area in which you enter the query is a multi-line, scrollable field, so you can format your SELECT statement any way you choose.

Full Support of SQL

SQL*ReportWriter lets you take full advantage of ORACLE's SQL support. Not only can you create queries using standard ANSI SQL, but you can take advantage of the additional power of unique ORACLE extensions as well: date, numeric, and string functions; outer joins and CONNECT BY clauses; and set operators such as UNION, INTERSECT, and MINUS.

Pick from a List of Values

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: Q_ORD						Tables		
SELECT Statement								
^								CUSTOMER
								DEPT
								EMP
v								ITEM
Parent-Child Relationships								
Parent Query 1:			Parent Query 2:					
	Child Columns		Parent 1 Columns		Pa			PRODUCT
^								SALES
v								
Choose a value from the list.								
Report Name: Order_Report						<List><Replace>		

List of Values

SQL*ReportWriter lets you pick table and column names from a pop-up List of values. Many other attribute settings also have a List of values containing all valid options for that setting. Whenever you see <List> at the bottom of a screen, you can press the [List of values] key to see the list. When you choose a value from the list for an attribute, it will automatically appear in the setting.

Enter Any SELECT Statement Directly

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Name: Q_ORD						Query 1 of 1		
SELECT Statement								
^	SELECT ORDID, SHIPDATE, CUSTID, TOTAL FROM ORD WHERE SHIPDATE > '31-DEC-86' ORDER BY ORDID							
V								
Parent-Child Relationships								
Parent Query 1:			Parent Query 2:					
	Child Columns	Parent 1 Columns			Parent 2 Columns			
^								
V								
Enter a SQL SELECT statement defining data for this report.								
Report Name: Order_Report						<List><Replace>		

The query above specifies information about customer orders from the ORD table. This is the only query needed for Order_Report. In Chapter 4, you will see how you can use multiple queries to create more complex reports.

Instant Validation Checks

SQL*ReportWriter immediately checks the SQL statement, looking up appropriate table and column definitions in the data dictionary, and validating the syntax. If there is an error, you will get a message indicating the type of problem. To clarify the message, just press the [Help] key to see a help screen for that message. Every SQL*ReportWriter screen contains validation checks to save you time while you build reports.

Sophisticated Defaults

Three simple steps are all you need to create a default report with SQL*ReportWriter name the report, name the query, and enter the SQL statement. With SQL*ReportWriter's default settings, most of the work is done for you. The column headings, column widths, and display formats are all automatically entered in your report definition, and your report is ready to run.

Executing the Report

Easy Prototyping

SQL*ReportWriter lets you check your progress at any time via its integrated report browser. Thus, you can see the effect of each change, which allows quick testing and easy prototyping. To see what Order_Repoti looks like with just the default settings, simply choose **Execute** from the **Action** Menu:

Create a Report
in One Minute

Ordid	Shipdate	Custid	Total
610	08-JAN-87	101	101.4
611	11-JAN-87	102	45
612	28-JAN-87	104	5860
613	01-FEB-87	108	6400
614	05-FEB-87	102	23940
615	06-FEB-87	107	710
616	10-FEB-87	103	764
617	03-MAR-87	105	46370
618	06-MAR-87	102	3510.5
619	04-MAR-87	104	1260
620	12-MAR-87	100	4450
621	01-JAN-87	100	730

Using the Help System

SQL*ReportWriter contains a comprehensive online help system that you can access at anytime during the development and maintenance of a report by pressing the [Help] key. Every menu choice, attribute setting, and error message has a help screen. The following screen is the help for the Query Screen.

Use the Help System for Complete Online Help

Help Menu

Highlighted Term

Bolded Term

Previous Next MainTopic Index Contents Example Quit
The Query Screen 1 of 1

The Query Screen lets you specify the SQL SELECT statement, a query name, and relationships for each query in the report. Most reports require only a single query; more than one query is required for reports containing information that is partially or completely disjoint. On this screen you specify:

- the SQL **SELECT statement**
- the query name
- parent query 1 and parent query 2, which are used to create master/detail relationships, and matrix reports
- Child query columns, parent 1 columns, and parent 2 columns, which are used to define the related data in a multi-query report.

For more information, see Managing Queries and Query Rules.

Go to the topic describing this concept or keyword.
Report Name: Order_Report <INSERT>

Easy Access

This context-sensitive help system provides multiple ways to access each piece of information, including an alphabetical index, a table of contents, and numerous cross-references. Notice in the screen above the highlighted term, "SELECT Statement", and the several bolded terms (indicated here by underscores.) You can learn more about any bolded term on a Help screen by simply placing the cursor on it so that it is highlighted and pressing the [Select] key. Another screen will appear displaying a definition of that term.

Each help screen also contains a menu line with a set of options that enable you to move through the help screens or go to another section in the help system.

Helpful Example Reports

In addition to the reference material, the help system also contains instructions for creating many reports, from a simple tabular report to a more complex master/ detail/ summary report. The sample reports are a useful guide for the beginning user, as well as effective templates for any user to copy.

Modifying the Default Report

While the SQL*ReportWriter default settings give you an excellent start, you can change any default to build just the report you need. In this section, you will see how to enhance the default tabular report by using the Group, Field, Summary, and Text Screens.

We will edit the column headings, change the format of the TOTAL column, summarize data in the TOTAL column, add a page header, and alter the spacing. After these modifications, Order_Report will look like this:

Modify a Report in a Few Easy Steps

- ❶ New Field Labels
- ❷ Display Formats
- ❸ Summary
- ❹ Page Header
- ❺ Spacing

❹ - ORDER SUMMARY -				
❶	❶	❶	❶	❷
Order ID	Shipdate	Customer	Total	❺
610	08-JAN-87	101	\$101.40	
611	11-JAN-87	102	\$45.00	
612	20-JAN-87	104	\$5860.00	
613	01-FEB-87	108	\$6400.00	
614	05-FEB-87	102	\$23940.00	
615	06-FEB-87	107	\$710.00	
616	10-FEB-87	103	\$764.00	
617	03-MAR-87	105	\$46370.00	
618	06-MAR-87	102	\$3510.00	
619	04-FEB-87	104	\$1260.00	
620	12-MAR-87	100	\$4450.00	
621	01-JAN-87	100	\$730.00	
Sum				❸

Step 1 Changing Data Appearance: Field Labels

First, to make the information in Summit's Order_Report more readable, we will override the default field labels.

In SQL*ReportWriter, each database column or expression from a query is referenced in the report as one field. By default, each field acquires all the attributes of its source column, including the width, datatype, and column headings. This report contains four fields: ORDID, SHIPDATE, CUSTID, and TOTAL.

You can easily override the defaults for these fields in the Field Screens. Just choose **Field** from the main menu and alter the default settings.

Change Field Display in the Field Screens

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Field Settings								
	Field Name	Source	Group	Label				
^	ORDID	ORDID	G_ORD	Ordid				
	SHIPDATE	SHIPDATE	G_ORD	Shipdate				
	CUSTID	CUSTID	G_ORD	Custid				
	TOTAL	TOTAL	G_ORD	Total				
v	[Redacted]							>
Enter a name for this field.								
Report Name: Order_Report < Replace >								

Spreadsheet-Style Layout

Notice that the spreadsheet-style layout makes it easy to see a large amount of information at once, so you can quickly find the setting you're looking for. All SQL*ReportWriter screens scroll both horizontally, to allow you to see more settings; and vertically, to display additional records. The Field Settings span three screens, as indicated by the size and position of the horizontal elevator towards the bottom of the screen.

The default fields are already in place in Field Screen One. The default field label comes from the corresponding column or expression in the query. To change the default, simply enter a new value.

Enter a New Value to Change Any Layout Attribute

• New Field Labels

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Field Settings								
	Field Name	Source	Group	Label				
^	ORDID	ORDID	G_ORD	Order ID ❶				
	SHIPDATE	SHIPDATE	G_ORD	Shipdate				
	CUSTID	CUSTID	G_ORD	Customer ❶				
	TOTAL	TOTAL	G_ORD	Total				
v								>
Enter the field label for this field.								
Report Name: Order_Report <Replace>								

Step 2
Changing Data Appearance:
Display Formats

SQL*ReportWriter supports many different types of display formats for both numbers and dates. For example, you can control the number of decimal places and the presence of a currency notation.

You can find the Display Format attribute in Field Screen Two. To cause the TOTAL field to appear with a dollar sign and two decimal places, changing all but one leading zero to blanks, enter \$ZZZZZ9.99. To see a list of all the Display Formats supported, press [Help].

Scroll through Layout Attributes

Display Format

Action	Query	Group	Field	Summary	Text	Reports	Parameter	Help
Field Settings								
	Field Name	Data Type	Field Width	Display Format	Relative Position	Lines Before	Spaces Before	
	ORDID	NUM	4					
	SHIPDATE	DATE	9					
	CUSTID	NUM	6					
	TOTAL	NUM	10	\$ZZZZZ9.99				
v								>
Enter the format mask (e.g. \$999.99)								
Report Name: Order_Report <Insert>								

National Language Support

SQL*ReportWriter provides you with National Language Support, which makes it very easy for you to generate reports in the native language of your users. For example, you can run a report, specifying the language as French, and SQL*ReportWriter will automatically change the dates and currency notations appropriately. Furthermore, SQL*ReportWriter supports 8-bit characters, enabling you to take full advantage- of your terminal and printer's capabilities to print a variety of national language characters.

Non-Procedural Interface

As you can see, SQL*ReportWriter does not require you to learn a new language. The logical user-interface lets you specify everything you need by typing into a form. You can spend your time designing reports instead of writing pages and pages of code.

You can use the Field Screens to specify how fields are displayed, such as changing the size and spacing of a field, altering its position, and adding anew field. In Chapters 3 and 4 we will demonstrate more uses of the Field Screens.

Step 3 Summarizing Data

We want to accumulate the total sales from all the orders into a grand total. To perform this calculation, we simply define a summary field in the Summary Screens.

After choosing **summary** from the SQL*ReportWriter main menu, you will see the following screen:

Create Summary Fields Here

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Summary Setting								
	Summary Name	Field	Function	Data Type	Width	Display Format		
^								
v								
								>
Enter a name for this summary field.								
Report Name: Order_Report <Replace>								

The screen is empty because summaries are not created by default. To summarize data in a field, just give the summary any meaningful name (SUM in this example), enter the field to be summarized (TOTAL), and specify the function SQL*ReportWriter should use. We will use the Sum function to add all values in the TOTAL field. This is all you need to create a summary field.

You can use the Summary Screens for more than simply adding values. There are fourteen different functions in these screens, including minimum, maximum, first, last, percent of total, count, and average. The **First** and **Last** functions enable you to print the first or last field value fetched from the database. Most of these functions also have a corresponding "running" function that shows cumulative results. These functions are provided in *addition* to over forty functions available as part of ORACLES extended SQL.

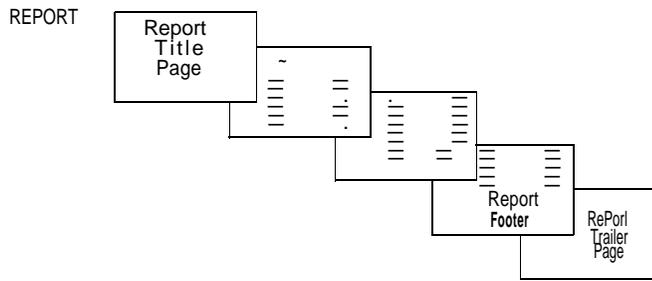
Summaries are very flexible in SQL*ReportWriter in that you can perform calculations on summary fields, and print totals for each page. You can also allow a summary field to be presented differently than the source field it summarizes by specifying a particular display format in the Summary Screens.

Change Summary Format

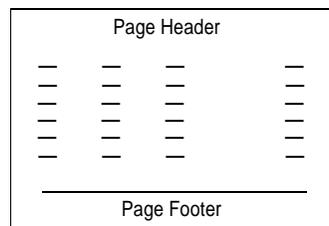
Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Summary Settings								
	Summary Name	Field	Function	Data Type	Width	Display Format		
^	Sum	Total	Sum	NUM	10	\$ZZZZZZ9		
v								>
Enter a name for this summary field.								
Report Name: Order_Report							< Insert >	

Step 4 Adding Text

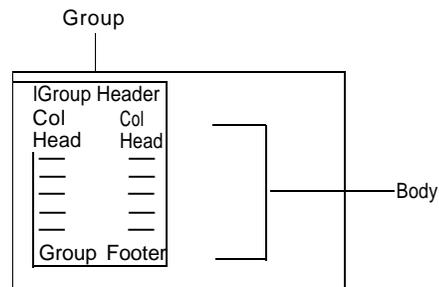
We have now specified the primary elements of Order_Report. However, adding a page header will give the report a more finished look. Page headers in SQL*ReportWriter are one of many kinds of text objects. Text objects are physical areas in a report filled with boilerplate text and references to fields, summaries, system variables, and



PAGE



GROUP



parameters. Text objects give you control over common repeated components such as headers, footers, and column headings, as well as title pages, trailer pages, and body text. Default versions of some text objects are generated by SQL*ReportWriter, while other text objects must be created manually.

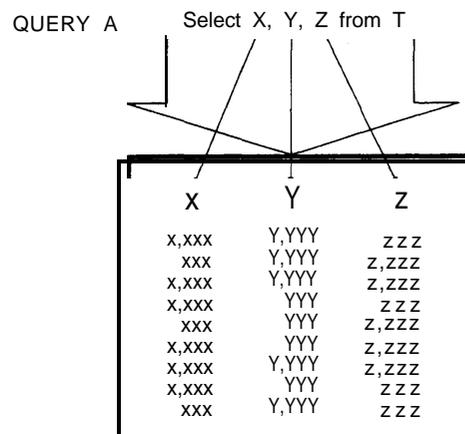
The figure above illustrates the flow of text objects in a report. Notice that you can manipulate text objects at many different levels. At the report level, you have text objects that appear at the beginning and/or end of a report, such as a report title page and header. Text objects at the page level appear on every page, and group text objects repeat for each specified group of fields. (We'll learn more about groups later.)

Scroll through the Text Settings screens until the text object displayed is "Page Header." To add a centered header to each page, set the Justification to "Center," and type in the text you want, beginning from the left side of the text area. The *SQL*ReportWriter Reference Manual* contains full information on various text objects available as well as their settings.

Step 5
Controlling
Report Layout
Group Spacing

We have now specified all the primary components of Summit's Order_Report the data, summary, and header. However, additional spacing around the fields and the header will make the report more readable.

One way to control the spacing around and between text and data is by manipulating *groups*. A group is a distinct set of fields. When you add a query to a report, SQL*ReportWriter creates one group and assigns all the fields from the query to it.



Groups are one of SQL*ReportWriter's most unique and powerful tools because they give you complete control over the layout of your data. In this report, we use groups for spacing. However, you can also use groups to position data, define control breaks, specify the direction in which data should print on the page, and for many other tasks. In the next chapter, you will see how to use groups to perform some of these other functions.

Group specifications span three Group Screens. After choosing Group from the main menu, you will see in Group Screen One that SQL*ReportWriter has automatically created the G_ORD group from the Q_ORD query.

Manipulate Groups of Fields in the Group Screens

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Group Settings								
	Group Name	Query	Print Direction	Matrix Group	Page Break			
^	G_ORD	Q_ORD	Down					
V								
								>
Enter a name for this group.								
Report Name: Order_Report							< Replace >	

To add an extra line between the page header and column headings, move to Group Screen Two and enter a "1" in Lines Before. Many other layout attributes also appear on this screen.

Add Spacing in Group Screen Two

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Group Settings								
	Group Name	Relative Position	Lines Before	Spaces Before	Spacing Record	Field	Fields Across	
^								
V	<							>
Enter the number of lines to leave blank before printing this group.								
Report Name: Order_Report							< Replace >	

After making these quick changes in the Field, Summary, Text, and Group Screens, the tabular Order_Report looks very different

The Modified Order_Report

Order ID	Shipdate	Customer	Total
610	08-JAN-87	101	\$101.40
611	11-JAN-87	102	\$45.00
612	20-JAN-87	104	\$5860.00
613	01-FEB-87	108	\$6400.00
614	05-FEB-87	102	\$23940.00
615	06-FEB-87	107	\$710.00
616	10-FEB-87	103	\$764.00
617	03-MAR-87	105	\$46370.00
618	06-MAR-87	102	\$3510.50
619	04-FEB-87	104	\$1260.00
620	12-MAR-87	100	\$4450.00
621	01-JAN-87	100	\$730.00
Sum			\$94140.90

Chapter Summary

With the development of Order_Report, you have seen how to create reports with SQL*ReportWriter. While this report is still relatively simple, it contains elements of a much more useful report. The steps that we took to produce it are very much like those you would take to produce any kind of report-query the data, manipulate fields, perform calculations, add text, and position groups.

Easy-to-Use Interface

Report development is simplified using SQL*ReportWriter's fill-in-the-form interface, which has:

List of Values: SQL*ReportWriter saves you time by providing you with a List of values for tables and columns, as well as for many attribute settings.

Instant Validation Checks: SQL*ReportWriter minimizes errors by automatically checking table and column definitions, and validating each entry you make.

Fully Integrated Browser With the integrated browser, you can immediately see the effect of each change on your report, allowing for easy prototyping and testing.

Productivity-Enhancing Features SQL*ReportWriter helps you get your reports started quickly, and allows you to make major changes easily by giving you:

Spreadsheet-Style Layout Forms: The spreadsheet-style layout scrolls both horizontally and vertically to give you access to large amounts of information at once.

Intelligent Defaults: With SQL*ReportWriter's automatic defaulting capability, you can build a report in one minute with three easy steps name the report, name the query, and enter a SELECT statement. SQL*ReportWriter automatically enters default column headings, column widths, display formats, and more.

No Fixed Text: The default settings help you get started, but all parts of a report can be edited. *You can easily* override any default, such as field labels and display formats, by replacing it with your own.

Online Help System

Just like a complete reference manual, but context-sensitive to your particular screen, SQL*ReportWriter's help system dramatically reduces learning time by providing you with:

Easy Access: The answer to any question you may have while learning SQL*ReportWriter is just one keystroke away in the context-sensitive help system. You can also locate all information, via the alphabetical index, table of contents, and numerous cross-references.

Example Reports: In addition to reference material, the help system includes a set of example reports for you to learn from or to copy.

Full Support of ORACLE's SQL

SQL*ReportWriter lets you take full advantage of ORACLE's SQL language, and is also compatible with the ANSI standard SQL and IBM's DB2. SQL gives you the benefits of being a:

Industry Standard Language: You will not need to learn a whole new language in order to *use* SQL*ReportWriter; SQL is the only language you need to know. Furthermore, you will be able to communicate with a variety of databases through ORACLE's distributed architecture.

Set-at-a-Time, Non-Procedural Language: Retrieving information from the database is easy: all you need to do is to specify what data you want, not how to get it. SQL's set-at-a-time approach means that a single relational command can retrieve, update, or delete multiple records stored in the database.

CHAPTER

3

FLEXIBLE REPORTING

In Chapter 2, you saw the few easy steps necessary to build a report with SQL*ReportWriter. In this chapter, we will dramatically change Order_Report to illustrate the flexibility and maintenance productivity of SQL*ReportWriter.

For each order, we will add the associated line item detail, creating a “master/ detail” report. We will then rearrange the field labels, add a field showing each line item as a percent of the total order, and vertically align the fields from the master group.

We will also show you how to change the relative position of groups in a report, and introduce you to SQL*ReportWriter’s powerful production reporting capabilities.

The Master/Detail Report

Our revised report will look like this

The New order_Report

-ORDER SUMMARY-						
Created Friday , the twelfth of May						
	Item	Product	Price	Qty	Amount	Percent
Order ID:	610					
Shipped:	08-JAN-87					
Customer:	101					
Total:			\$101.40			
	1	100860	35.00	1	35.00	34.52%
	2	100870	2.80	3	8.40	8.28%
	3	100890	58.00	1	58.00	57.20%
Order ID:	611					
Shipped:	11-JAN-87					
Customer:	102					
Total:			\$45.00			
	1	100861	45.00	1	45.00	100.00%
order ID:	612					
Shipped:	20-JAN-87					
Customer:	104					
Total:			\$5860.00			
	1	100860	30.00	100	3000.00	51.19%
	2	100861	40.50	20	810.00	13.82%
	3	101863	10.00	150	1500.00	25.60%
	4	100871	5.50	100	550.00	9.39%

While in most report writers these changes could take hours, in SQL*ReportWriter they take only minutes. You will now see how to produce a report like this starting with the version of Order_Report shown on page 2-18.

Queries With JOINS

The first version of Order_Report lists all the orders taken by Summit after 1986. To add data reflecting the products ("line items") sold with each order, we will alter the query to include data from the ITEM table by specifying a JOIN condition. Note that a JOIN is necessary only because the data in our report is stored in two tables instead of one. Another way to build this report would be to add a second query and let SQL*ReportWriter combine the data in the two tables. SQL*ReportWriter's ability to incorporate multiple queries in a report is very powerful because many reports require data that cannot be accessed with just one query—even if you specify JOIN conditions. We will see examples of multi-query reports in Chapter 4.

Recall that the first query looks like the following

Original Query
For Order_Report

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: Q_ORD						Query 1 of 1		
^	SELECT Statement							
	SELECT ORDID, SHIPDATE, CUSTID, TOTAL FROM ORD WHERE SHIPDATE > '31-DEC-86' ORDER BY ORDID							
V								
Parent-Child Relationships								
Parent Query 1:			Parent Query 2:					
	Child Columns	Parent 1 Columns			Parent 2 Columns			
^								
V								
Enter aSQL SELECT statement defining data for this report.								
Report Name: Order_Report						<List><Replace>		

We will change the query to select columns from the ITEM table as well. Notice that we JOIN the data from these two tables by specifying that ORDID in the ORD table match ORDID in the ITEM table.

Alter Query to Select Data
From Item Table

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: Q_ORD						Query 1 of 1		
^	SELECT Statement							
	SELECT ORD.ORDID, SHIPDATE, CUSTID, TOTAL, ITEMID, PRODID, ACTUAPRICE, QTY, ITEMTOT FROM ORD, ITEM WHERE ORD, ORDID = ITEM.ORDID AND SHIPDATE > '31-DEC-86' ORDER BY ORD.ORDID, ITEMID							
V								
Parent-Child Relationships								
Parent Query 1:			Parent Query 2:					
	Child Columns	Parent 1 Columns			Parent 2 Columns			
^								
V								
Enter a SQL SELECT statement defining data for this report.								
Report Name: Order_Report						<List><Replace>		

Section-at-a-Time Layout

SQL*ReportWriter creates a group called G_ORD for this query. Next, we create a separate section of the report to display the item-level information by creating another group called G_ITEM. Go to Group Screen One and insert G_ITEM below G_ORD.

Create a New Group

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Group Settings								
		Group Name	Query	Print Direction	Matrix Group	Page Break		
^								
v								
								>
Enter a name for this group.								
Report Name: Order_Report							<Replace>	

Each group represents a section or subsection of a report. Using the group positioning controls, you can place groups anywhere on the page, moving all the fields in the group together. You can put one group below another, or place them side-by-side.

We want to insert two blank lines between the data for each order. This can be accomplished by changing the default value for Record Spacing on Group Screen Two.

Add Spacing Between Each Record

Two Lines between each G_ORD Record

Action Query Group Field Summary Text Report Parameter Help						
Group Settings						
	Group Name	Relative Position	Lines Before	Spacing Record	Field	Fields Across
^	G_ORD G_ITEM		1	2		
V						

< [REDACTED] >

Choose the position of this group in relation to the previous group.

Report Name: Order_Report <Replace>

We also need to move the field labels in G_ORD from their default position as column headings, positioning them next to each field value. We do this by changing the Label Position from "Above" (default) to "Left".

Change Label Positioning

G_ORD's New Label Position

Action Query Group Field Summary Text Report Parameter Help						
Group Settings						
	Group Name	Multi-Panel	Label Position	Field	Highlight Label	
^	G-ORD G-ITEM		Left			
V						

< [REDACTED] >

Keep all fields in a record on the same panel.

Report Name: Order_Report <Replace>

More Calculations

In this section, you will see how to expand the information about Summit's orders and items using SQL*ReportWriter's ability to calculate values from the retrieved data.

SQL*ReportWriter's calculation capabilities extend beyond the simple grand total you saw in Chapter 2. In this chapter, we will create a computed field that shows each item as a percent of the total order to illustrate some of the powerful calculations possible with SQL*ReportWriter.

A computed field is a completely new field created by performing a calculation on an existing field. It behaves like any other field except that it is not created by default.

You can create a computed field in the Field Screens by inserting a new record and entering the appropriate information. Note that fields appear in your report in the order they appear on this screen. You can rearrange fields by first deleting the field, moving the cursor, then undeleting the field.

Create a Computed Field

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Field Settings								
	Field Name	Source	Group	Label				
^	ORDID	Q_ORD.ORDID	G_ORD	Order ID				
	SHIPDATE	SHIPDATE	G_ORD	Shipped				
	CUSTID	CUSTID	G_ORD	Customer				
	TOTAL	TOTAL	G_ORD	Total				
	ORDID2	Q_ITEM.ORDID	G_ITEM	Ordid				
	ITEMID	ITEMID	G_ITEM	Item				
	PRODID	PRODID	G_ITEM	Product				
	ACTUALPRICE	ACTUALPRICE	G_ITEM	Price				
	QTY	QTY	G_ITEM	Quantity				
	ITEMTOT	ITEMTOT	G_ITEM	Amount				
v	PCT_TOTAL	ITEMTOT	G_ITEM	Percent				
	Enter a name for this field.							<
Report Name: Order_Report							< Replace >	

First, name the new field and then identify the column it is based on. We will name it "PCT_TOTAL", and the column from which PCT_TOTAL will be calculated is ITEMTOT.

Next, assign the new field to a group, and specify its field label. Because we want to calculate a value for each line item, we enter "G_ITEM" as PCT_TOTAL's group, and enter "PERCENT" as its field label.

The only other essential information for a computed field is on Field Screen Three. On this screen, indicate the function to use and the reset group. The reset group tells SQL*ReportWriter how frequently the new field should be reset to zero. The field will be reset every time SQL*ReportWriter encounters the end of the specified group.

The function to calculate the proper value of PCT_TOTAL from ITEMTOT is % Total, which computes the percent of the total contributed by each item. We choose a Reset Group of G_ORD to cause the percentage to be based on each order rather than the report as a whole.

Specify the Function and Reset Level

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Field Settings								
	Field Name	Align	Skip	Repeat	Computed Value		Function	Reset Group
^	ORDID SHIPDATE CUSTID TOTAL ORDID2 ITEMID PRODID ACTUALPRICE QTY ITEMTOT							
V	PCT_TOTAL					% Total	G_ORD	
<								
Choose the justification for this field.								
Report Name: Order_Report							<List><Replace>	

The same function/preset group concepts apply to summaries. In fact, the only difference between computed fields and summaries is that summaries appear after the data being summarized, while computed fields appear as one of the fields in the group.

Fine-Tuning the Reperk Fields

To make Order_Report more readable, change the field labels, add display formats, realign the G_ORD fields, and edit text in the G_ORD group.

Just as we changed the field labels in the G_ORD group in Chapter 2, we will now edit the field labels in the G_ITEM group for clarity, as shown. Notice that we have also changed the Group of all fields selected from the ITEM table from G_ORD to G_ITEM.

Change Field Labels on Field Screen One

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help	
Field Settings									
	Field Name	Source		Group		Label			
^	ORDID	ORDID		G_ORD		Order ID			
	SHIPDATE	SHIPDATE		G_ORD		Shipped			
	C U S T I D	CUSTID		G_ORD		Customer			
	TOTAL	TOTAL		G_ORD		Total			
	ITEMID	ITEMID		G_ITEM		Item			
	PRODID	PRODID		G_ITEM		Product			
	ACTUALPRICE	ACTUALPRICE		G_ITEM		Price			
	QTY	QTY		G_ITEM		Qty			
	ITEMTOT	ITEMTOT		G_ITEM		Amount			
	PCT_TOTAL	ITEMTOT		G_ITEM		Percent			
v	CURRDATE	&DATE		G_ITEM		Currdate			
	Enter a name this field.							>	
Report Name: Order_Report							<Replace>		

We will also add some display formats. In addition to the formatting shown in this example, SQL*ReportWriter supports formats such as parentheses, debit/credit notation, phone numbers, social security numbers, fixed and floating currency symbols, and a wide variety of date formats.

System Variables

SQL*ReportWriter provides you with the &DATE, &PAGE, and &NUM_PAGES system variables for convenience in printing dates, numbering the pages, and printing the total number of pages in your report. You can date your report with the current date, for example, simply by placing the system variable &DATE wherever you wish in the Text Screens. However, we want a different format than the default DD-MON-YY, so we will create a new field, CURRDATE, as shown in the Field Screen above. CURRDATE has the &DATE system variable as its source, REPORT as its group, and a format of: fmDay, "the" ddspth "of" fmMonth. This will spell out the current date without leading blanks, like "Monday, the twelfth of June".

Change Display Formats

Action Query Group Field Summary Text Report Parameter Help							
Field Settings							
	Field Name	Data Type	Field Width	Display Format	Relative Position	Lines Before	Spaces Before
^	ORDID	NUM	4				
	SHIPDATE	DATE	9				
	CUSTID	NUM	6				
	TOTAL	NUM	8	\$ZZZZ9.93			
	ITEMID	NUM	3				
	PRODID	NUM	6				
	ACTUALPRICE	NUM	8	ZZZZ9.99			
	QTY	NUM	3				
	ITEMTOT	NUM	8	ZZZZ9.99			
	PCT_TOTAL	NUM	10	ZZ9.99%			
v	CURRDATE	DATE	35	fmDay, "t			
		<				>	
Enter the width of this field in spaces.							
Report Name: Order_Report <Replace>							

Finally, in the Field Screens, we will align the numeric fields in G_ORD to the left (by default, they are aligned to the right), and skip the unnecessary field, ORDID2, which already appears in G_ORD.

Align G_ORD Fields

Action Query Group Field Summary Text Report Parameter Help							
Field Settings							
	Field Name	Align	Skip	Repeat	Computed Value		
					Function	Reset	Group
^	ORDID	Left					
	SHIPDATE						
	CUSTID	Left					
	TOTAL	Left	X				
	ORDID2						
	ITEMID						
	PRODID						
	ACTUALPRICE						
	QTY						
	ITEMTOT						
	PCT_TOTAL				%Total		G_ORD
v	CURRDATE						
		<				>	
Choose the justification for this field.							
Report Name: Order_Report <List><Insert>							

These changes take only seconds of your time and will make the data fit perfectly into your report specifications.

First, we will add "Created: &CURRDATE" to the Page Header of Order_Report. Then, we will restructure the body text of G_ORD to make all the data better fit on one page. Move to the Text Screens and scroll through the text objects until you get to "G_ORD Body". Note that the fields are referenced in text objects by placing an ampersand (&) before the field name.

Edit G_ORD's Default
Body Text

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
				Text	Settings			
			Order ID:		&ORDID			
			Shipped:		&SHIPDATE			
			Customer:		&CUSTID			
			T o t a l :		& T O T A L			

Edit the text.

Report Name: Order_Report <Normal> <Insert>

We will stack the fields in G_ORD on top of each other by inserting carriage returns before each field, and line them up to leave more space on the page.

After making these changes to the first version of Order_Report, it now looks very different

The Final Order_Report

-ORDER SUMMARY-							
Created Friday, the twelfth of May							
	Item	Product	Price	Qty	Amount	Percent	
Order ID:	610	1	100860	35.00	1	35.00	34.5%
Shipped:	08-JAN-87	2	100870	2.80	3	8.40	8.28%
Customer:	101	3	100890	58.00	1	58.00	57.28%
Total:	\$101.40						
Order ID:	611	1	100861	45.00	1	45.00	100.00%
Shipped:	11-JAN-87						
Customer:	102						
Total:	\$45.00						
Order ID:	612	1	100860	30.00	100	3000.00	51.19%
Shipped:	20-JAN-87	2	100861	40.50	20	810.00	13.8%
Customer:	104	3	101863	10.00	150	1500.00	25.60%
Total:	\$5860.00	4	100871	5.50	100	550.00	9.39%

Using SQL*ReportWriter, you can make major changes very quickly, and you have total control over every aspect of your report. It would take you hours of work and pages of programming to create this report with a traditional reporting tool. With SQL*ReportWriter you get improved productivity without sacrificing any report formatting power.



Production Reporting

Once your report is defined, you can run the report to run as often as you wish for different sets of data. SQL*ReportWriter provides several features to give you the control necessary for “production reporting”, including a dedicated runtime, or “batch”, engine; printer drivers; an interactive browser; the ability to embed parameters in your reports; and a Run-time Parameter Form.

Batch Reporting

SQL*ReportWriter has a command line interface, which enables you to execute reports from the operating system command line. Using the RUNREP command, you can specify what report to run, how many copies, parameter values, and when and where to print without any other interaction.

Printer Drivers

SQL*ReportWriter provides drivers for several common printers. In addition, you can create drivers for other printers with a simple utility. SQL*ReportWriter supports several highlighting modes, such as bold and underline. Furthermore, you can define and use up to 32,000 general purpose printer codes to take advantage of special functions on your printer, or to specify graphic characters.

Interactive Browser

If your report output is longer than one page, you can use SQL*ReportWriter’s interactive browser to examine it page by page. You can view your entire report by scrolling forwards, backwards, and side-to-side.

Embedded Parameters

SQL*ReportWriter lets you embed parameters in your report to specify boilerplate text and data selection criteria. Parameters can be referenced in text objects and in queries.

You can specify parameter values as arguments directly on the command line or enter them into the Run-time Parameter Form.

We will specify a parameter that tells SQL*ReportWriter how to sort the items in each customer order. First enter the new parameter, ORD.CRITERIA, in the Parameter Settings screen. We will specify a default value of ITEMID.

Enter New Parameter in Parameter Settings Screen

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Parameter Settings								
	Parameter Name	Data Type	Width	Default Value	Label			
^	DESTYPE	CHAR	80	Screen	Destination type			
	DESNAME	CHAR	80		File Name / Spool Dev			
	DESFORMAT	CHAR	80	dfit	Printer Description F			
	COPIES	NUM	2	1	Number of Copies			
	CURRENCY	CHAR	4		Currency Symbol			
	THOUSANDS	CHAR	1		Thousands Separator			
	DECIMAL	CHAR	1		Decimal Indicator			
	ORD_CRITERIA	CHAR	40	ITEMID	Ord Criteria			
v								>
Enter a name for this parameter.								
Report Name: Order_Report							<Replace>	

Then we go to the Query Settings screen, and insert ORD_CRITERIA in place of ITEMID in the query Q_ORD.

Enter New Parameter in Query

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: Q_ORD			SELECT Statement			Query 1 of 1^		
^	SELECT ORD.ORDID, SHIPDATE, CUSTID, TOTAL ITEMID, PRODDID, ACTUALPRICE, QTY, ITEMTOT FROM ORD, ITEM WHERE ORD, ORDID = ITEM, ORDID AND SHIDATE > '31-DEC-86'							
v	ORDER BY ORD.ORDID, &ORD_CRITERIA							
Parent-Child Relationships								
Parent Query 1:			Parent Query 2:					
	Child Columns	Parent 1 Columns	Parent 2 Columns					
^								
v								
Enter a SQL SELECT statement defining data for this report.								
Report Name: Order_Report							<List><Replace>	

Run-time Parameter Form

This form allows you to override Print options and parameter values just before you run your report. Furthermore, if you do not want the Run-time Parameter Form to appear at all, you can easily suppress the display by indicating this when starting SQL*ReportWriter. Parameter values can be supplied on the command line when you invoke SQL*ReportWriter from your operating system.

The Run-time Parameter form can be customized according to your needs. For instance, you can give it a more descriptive title than the default value of "Parameter Values" such as "Printer Information." The hint line and status lines may also be changed to give the user more information about the parameters than the defaults allow. These changes are made in the Report Settings Screen. The following screen is an example of a customized Run-time Parameter Form, and it includes the sort condition we entered for our report.

Now, the data sort criteria can be specified each time you run the report.

Default Sort Criteria in Run-time Parameter Form

Printer Information	
Parameter	Value
Destination Type	Screen
File Name / Spool Device	Order_Report.lis
Printer Description File	dflt
Number of Copies	1
Ord Criteria	ITEMID

Enter the desired printer values.

Report is printed in room 219B <Replace>

We will change the sort criteria in the report by just typing in one or more values on the Run-time Parameter Form; for example, we can sort by QTY and ACTUALPRICE instead.

Enter New Parameter in Run-time Parameter Form

Printer Information	
Parameter	Value
Destination Type	Screen
File Name / Spool Device	Order_Report.lis
Printer Description File	dflt
Number of Copies	1
Ord Criteria	QTY, ACTUALPRICE

Enter the desired printer values.

Report is printed in room 219B. <Replace>

And here is Order_Report with the data sorted by QTY and ACTUAL.PRICE instead of by ITEMID.

Order Report Sorted by QTY and ACTUALPRICE

- ORDER SUMMARY -							
Created Friday, the twelfth of May							
	Item	Product	Price	Qty	Amount	Percent	
Order ID:	610	1	100060	35.00	1	35.00	34.52%
Shipped:	08-JAN-87	3	100090	58.00	1	58.00	57.28%
Customer:	101	2	100070	2.80	3	8.40	8.20%
Total:						\$101.40	
Order ID:	611	1	100061	45.00	1	45.00	100.00%
Shipped:	11-JAN-87						
Customer:	102						
Total:						\$45.00	
Order ID:	612	2	100061	40.50	20	810.00	13.82%
Shipped:	28-JAN-87	4	100071	5.50	100	550.00	9.35%
Customer:	104	1	100060	30.00	100	3000.00	51.19%
Total:		3	101063	10.00	150	1500.00	25.68%



Chapter Summary

By observing a relatively simple default report transform into a complete master/ detail report you have seen how quickly you can make major changes using SQL*ReportWriter. In this chapter, we created anew group, G_ITEM, and assigned the “detail” fields to it to break our default report into a master/detail report. We also demonstrated more advanced uses of the SQL*ReportWriter screens to illustrate powerful capabilities, such as:

Section-at-a-Time Layout

You can set up your report layout section by section. In Order_Report, we retained the default placement for groups, which placed G_ITEM to the right of G_ORD. We also added two lines between the records within the group G_ORD, and changed G_ORD’s label positioning.

Powerful Calculation Capabilities

You can quickly perform useful calculations with your data. Fourteen functions — including running functions – are available to create new fields (and summaries.) We created a computed field in Order_Report to show each item as a percent of the total order for each customer.

Fine-Tuning Your Report

In addition to defining the data and overall appearance of your report, you can “fine-tune” it. In this chapter, we used the field screens to change field labels, add display formats including anew date format, and align G_ORD’s numeric fields to the left. We also edited G_ORD’s body text to flow vertically instead of horizontally.

Production Reporting

SQL*ReportWriter lets you control report production with batch reporting, printer drivers, embedded parameters, and a customizable Run-time Parameter Form.

CHAPTER

4

POWERFUL REPORTING

You have now seen how to use SQL*ReportWriter to develop and maintain a default and a master/detail report. In this chapter, we will show you additional ways to tap the power of SQL*ReportWriter with examples of a variety of commonly used reports. Of course, there are no rigid report types; you can build any type of report you want.

The examples in this chapter include a matrix report, a form letter, mailing labels, and a more complex master/detail/summary report. Because these are relatively detailed examples, we suggest that you either spend some time studying each example thoroughly, or simply glance through them to see the power of SQL*ReportWriter.

The Matrix Report

TOTAL ORDERS BY CUSTOMER						
Cust.id	86/85	86/86	86/87	86/88	87/01	87/02
100			\$3	\$98	\$738	
101					\$181	
102		\$288			\$45	\$23948
103						\$764
104			\$41		\$5868	\$1268
105						
106	\$2	\$698	\$8324			
107						\$718
108						\$6488

This matrix report contains the total orders for each customer with one column for each month. The following screens show the necessary settings for this matrix report.

The Queries

A matrix report contains a cross-tabulation of information created with three queries. The following screens show the queries in the order they should be created. Notice the parent/child relationships and matching columns among the three queries.

Define the Rows

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: Q_CUST						Query 1 of 3		
SELECT Statement								
^	SELECT DISTINCT CUSTID							
	FROM ORD							
v								
Parent_Child Relationships								
Parent Query 1:				Parent Query 2:				
	Child Columns	Parent 1 Columns	Parent 2 Columns					
^								
v								
Enter a name for this query								
Report Name: Matrix_Report							<Replace>	

Q_CUST retrieves the customer information, which define the matrix rows. The second query, Q_MONTH, retrieves the months, which define the columns of the matrix.

Define the Columns

Query Settings								
Query Name: Q_Month						Query 2 of 3		
SELECT Statement								
^	SELECT DISTINCT CUSTID							
	FROM ORD							
v								
Parent_Child Relationships								
Parent Query 1:				Parent Query 2:				
	Child Columns	Parent 1 Columns	Parent 2 Columns					
^								
v								
Enter a name for this query.								
Report Name: Matrix_Report							<Replace>	

Define the Cells

① Two Parent Queries

② Matching Columns

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: Q_CELL		SELECT Settings				Query 3 of 3		
^	SELECT TO_CHAR(SHIPDATE, 'YY/MM') MONTH, CUSTID, SUM(TOTAL) TOTAL FROM ORD GROUP BY TO_CHAR(SHIPDATE, 'YY/MM'), CUSTID ORDER BY TO_CHAR(SHIPDATE, 'YY/MM'), CUSTID							
v								
Parent-Child Relationship								
①	Parent Query 1: Q_CUST		Parent Query 2: Q_MONTH					
	Child Columns	Parent 1 Columns	Parent 2 Columns					
^	CUSTID ② MONTH ②	CUSTID	MONTH					
v								
Enter a name for this query								
Report Name: Matrix_Report							<Replace>	

Finally, Q_CELL retrieves the total orders aggregated by customer and month.

Group Settings

SQL*ReportWriter creates one group for each query. You need to override the defaults for Print Direction and Matrix Group, as shown. Specify the Print Direction for G_CUST as Down and G_MONTH as Across, so the months print in a row, and the customers print in a column. G_CELL has a Crosstab Print Direction so that it will fill in the cells of the matrix. The Matrix Group setting defines the groups as part of a matrix.

Define Three Matrix Groups

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Group Settings								
	Group Name	Query	Print Direction	Matrix Group	Page Break			
^	G_CUST	Q_CUST	Down	X				
	G_MONTH	Q_MONTH	Across	X				
v	G_CELL	Q_CELL	Crosstab	X				
	Enter a name for this group.							>
Report Name: Matrix_Report							<Replace>	

Print Directions

Matrix Groups

Next, move to Field Screen Two to change the field widths of MONTH and TOTAL to 6, and to add the display format, \$ZZZZZ9, to TOTAL. In Field Screen Three, place an "X" in the Skip column for the redundant MONTH2 and CUSTID2 fields to prevent them from printing.

Suppress Identical Fields

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Field Settings								
	Field Name	Align	Skip	Repeat	Computed Value			
					Function	Reset	Group	
^	CUSTID							
	MONTH							
	MONTH2		X					
	CUSTID2		X					
	TOTAL							
V								
	<							
Choose the justification for this field.								
Report Name: Matrix_Report							<List><Replace>	

Finally, we will make a few changes in the Text Screens to make the report more readable. Delete the column headings for G_MONTH and G_CELL, and add dashes beneath &MONTH in the body of G_MONTH. Then, add a couple of spaces before &MONTH and &TOTAL in the body of G_MONTH and the body of G_CELL. This is all that's required to create a matrix report with SQL*ReportWriter.

The Form Letter Report

Name: NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER
Account: 108
Address: 98 LONE PINE WAY
HIBBING, MN 55649

Credit Limit: \$8000.00

Dear Sir:

In compliance with company policy, we are confirming the your current credit limit as \$8000.00. If this conflicts with your records, or if your company is no longer located at 98 LONE PINE WAY, HIBBING, MN 55649, please notify Customer Service as soon as possible.

Sincerely,

John J. Smith

In a form letter report, you can mix your own text with data retrieved from the database. SQL*ReportWriter will automatically wrap each copy of the letter on word boundaries after it retrieves the data. In this report, the data has been placed in a form that verifies the credit limit and address of customers.

The following screens show the settings necessary to build this report.

The Query

The query for this report retrieves all the data needed for the form letter: the name, customer identification number, address, city, state, zip code, and credit limit.

Query the Customer Table

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
	Query Name: Q_CUST					Query 1 of 1		
^	SELECT Statement							
	SELECT NAME, CUSTID, ADDRESS, CITY, STATE, ZIP, CREDIBILITY FROM CUSTOMER ORDER BY ZIP							
v								
Parent_Child Relationships								
	Parent Query 1:				Parent Query 2:			
	Child Columns	Parent 1 Columns	Parent 2 Columns					
^								
v								
Enter a name for this query.								
Report Name: Form_Letter							<Replace>	

Group Settings

After specifying the query, go to Group Screen One and enter "Always" as the Page Break attribute. This setting forces a page break after each record, so the data for only a single customer appears on each page. This causes SQL*ReportWriter to print one letter per page.

Limit to One Customer Per Page

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
	Group Name	Query	Print Direction	Matrix Group	Page Break			
^								
v								
Enter a name for this group.								
Report Name: Form_letter							<Replace>	

Field Settings

Next, use Field Screen Two to adjust the field widths to fit the form letter specifications, and set the display format of CREDITLIMIT to \$ZZZZZ9.99

Adjust Field Widths

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Field Settings								
	Field Name	Data Type	Field Width	Display Format	Relative Position	Lines Before	Spaces Before	
^	NAME	CHAR	45					
	CUSTID	NUM	3					
	ADDRESS	CHAR	30					
	CITY	CHAR	12					
	STATE	CHAR	2					
	ZIP	CHAR	5					
	CREDITLIMIT	NUM	9	\$ZZZZZ9.9				
v								
<								>
Enter the width of this field in spaces.								
Report Name: Form_Letter							<Replace>	

We will set the alignment of ADDRESS, CITY, and CREDITLIMIT to Variable in Field Screen Three so that there will be no unnecessary blank spaces when printing these values. This setting also causes SQL*ReportWriter to wrap the contents of each letter on word boundaries just before it is printed. Field alignment allows you to center, right-, and left-justify fields as well.

Text Edits

The last step is to clear unnecessary column headings and to edit the body text in the Text Screen. Using the Zoom function if needed, edit the text as shown, referencing fields with an ampersand (&). Because we have fields which are specified to be Variable, SQL*ReportWriter will try to word-wrap all of the text in G_CUST into a single paragraph. Therefore, we must place "hard returns", indicated by &CRs, in the text wherever we want to force a carriage return. Your body text can be as much as ten pages wide, and as long as needed.

Edit the Text

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Name:	&NAME	&CR						
Account:	&ACCTID	&CR						
Address:	&ADDRESS	&CR						
	&CITY, &STATE	&ZIP	&CR					
Credit Limit:	&CREDITLIMIT	&CR						
Dear Sir:	&CR							
In compliance with company policy, we are confirming that your current credit limit is &CREDITLIMIT. If this conflicts with your records, or if your company is no longer located at &ADDRESS, &CITY, &STATE &ZIP, please notify Customer Service as soon as possible. &CR								
Sincerely,	&CR							
John J. Smith								
Edit the text:								
Report Name:	Form_Letter						<Normal>	<Replace>

The Mailing Label Report

NORTH WOODS HEALTH AND FITNESS 98 LONE PINE WAY HIBBING, MN 55649	SHAPE UP 908 SEQUOIA PALO ALTO, CA 94301
K + T SPORTS 3476 EL PASEO SANTA CLARA, CA 91003	VOLLYRITE 9722 HAMILTON BURLINGAME, CA 95133
EVERY MOUNTAIN 574 SURRY RD, CUPERTINO, CA 93301	JOCKSPORTS 345 VIEWRIDGE BELMONT, CA 96711
WOMENS SPORTS VALCO VILLAGE SUNNYVALE, CA 93301	JUST TENNIS HILLVIEW HALL BURLINGAME, CA 97544
TKB SPORT SHOP 490 BOLI RD. REDWOOD CITY, CA 94061	

The Query

With SQL*ReportWriter, you can easily print mailing labels in multiple columns on each page. After entering the query, just make some simple positioning adjustments. First, retrieve the data with a SELECT statement

Select the Name and Address

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: Q_CUST						Query 1 of 1		
SELECT Statement								
^	SELECT NAME, ADDRESS, CITY, STATE, ZIP							
	FROM CUSTOMER							
	ORDER BY ZIP							
v								
Parent_Child Relationships								
Parent Query 1:				Parent Query 2:				
	Child Columns		Parent 1 Columns		Parent 2 Columns			
^								
v								
Enter a name for this query.								
Report Name: Mailing_Label								<Replace>

Group Settings

Set the Print Direction to Down/Across to allow the mailing labels to print down the length of the page until the bottom is reached. Then, SQL*ReportWriter will begin print the labels to the right of the topmost label if there is enough room on the page. We have changed the length of the report to 25 in the Report Screen to demonstrate Down/Across printing properly on your terminal display.

Set Print Direction

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Group Settings								
	Group Name	Query	Print Direction	Matrix Group	Page Break			
^	G_CUST	Q_CUST	Down/Across					
v								
Enter a name for this group.								
Report Name: Mailing_Label <Replace>								

Field Settings

Next, modify the field widths in Field Screen Two to fit the exact dimensions of the labels. To make this report, you should specify field widths as 30,30,12,2, and 5 for NAME, ADDRESS, CITY, STATE, and ZIP, respectively. Make the alignment of CITY in Field Screen Three Variable to avoid printing extra blank spaces.

Text Edits

The last step in building a mailing label report is to clear the column headings and edit the body text to fit your specifications. In order to get a line between each record, you must insert two blank lines in the text screen above the edited body text. Also insert 5 blank spaces before the body text for readability. Again, remember that because we have a variable field, we must place &CRs in the text wherever we want carriage returns.

The Master/Detail/Summary Report

An example of a more complex report that can be created with SQL*ReportWriter is the master/detail/summary report:

Customer	Product	Date	Cost
WOMENDS SPORTS	ACE TENNIS BALLS-3 PACK	06-FEB-87	280.00
	ACE TENNIS BALLS-6 PACK	06-FEB-87	250.00
	ACE TENNIS RACKET II	06-FEB-87	180.00
Customer Product Totals			
	ACE TENNIS BALLS-3 PACK		280.00
	ACE TENNIS BALLS-6 PACK		250.00
	ACE TENNIS RACKET II		180.00
First sporting goods stored geared exclusively towards woemen. Unusual promotion al style and very willing to take chances towards new product			
NORTH WOODS	ACE TENNIS BALLS-6 PACK	01-FEB-87	560.00
HEALTH AND	SB ENERGY BAR-6 PACK	01-FEB-87	440.00
FITNESS SUPPLY	SB VITA SNACK-6 PACK	01-FEB-87	600.00
CENTER	SP TENNIS RACKET	01-FEB-87	4,800.00

This report shows shipments to each customer, sorted by product and shipment date. The totals by product are listed below each customer record. Notice that the product totals are very different from a simple subtotal and they are shown as a separate sub-section of the report.

To create this type of report, you must use three queries that specify data from four tables: ORD, ITEM, CUSTOMER, and PRODUCT. Within the SELECT statements, you must query the columns needed for the report as well as the columns that join the tables together.

Four groups are built from the data in the three queries. One group has customer information, another has shipment information, a third has product summary information, and the last contains product information. Each group has been automatically created by SQL*ReportWriter as the result of a query except for the last (G_PROD), which is manually created.

The following screens illustrate how to build this master/detail/summary report.

The Queries

This report requires three SELECT statements:

The Master Query

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: Q_CUST						Query 1 of 3		
SELECT Statement								
^	SELECT CUSTID, NAME, COMMENTS							
	FROM CUSTOMER							
	WHERE CUSTID IN (107, 108)							
V	ORDER BY CUSTID							
Parent-Child Relationships								
Parent Query 1: Q_CUST				Parent Query 2:				
	Child Columns		Parent 1 Columns		Parent 2 Columns			
^	CUSTID		CUSTID					
V								
Enter a name for this query.								
Report Name: Master_Detail_Summary_Report							<Replace>	

The Detail Query

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: Q_SHIPMENT						Query 2 of 3		
SELECT Statement								
^	SELECT CUSTID, SHIPDATE, DESCRIP, ITEMTOT							
	FROM ORD, PRODUCT, ITEM							
	WHERE ITEM.ORDID = ORD.ORDID							
	AND ITEM.PRODID = PRODUCT.PRODID							
V	ORDER BY CUSTID, DESCRIP, ORDERDATE							
Parent-Child Relationships								
Parent Query 1: Q_CUST				Parent Query 2:				
	Child Columns		Parent 1 Columns		Parent 2 columns			
^	CUSTID		CUSTID					
V								
Enter a name for this query.								
Report Name: Master_Detail_Summary_Report							<Replace>	

The Summary Query

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: Q_SUMMARY						Query 3 of 3		
^	SELECT Statement							
	SELECT CUSTID, DESCRIP, SUM(ITEMTOT) TOTAL							
	FROM ORD, PRODUCT, ITEM							
	WHERE ITEM.ORDID = ORD.ORDID							
	AND ITEM.PRODID = PRODUCT.PRODID							
v	GROUP BY CUSTID, DESCRIP							
	ORDER BY CUSTID, DESCRIP							
Parent_Child Relationships								
Parent Query 1: Q_CUST						Parent Query 2:		
^	Child Columns	Parent 1 Columns		Parent 2 Columns				
	CUSTID	CUSTID						
v								
Enter a name for this query.								
Report Name: Master_Detail_Summary_Report							<Replace>	

Group Settings

Next, establish control breaks for each product within each customer. Instead of adding a fourth query, do this by creating a new group based on Q_SHIPMENT. Name the new group G_PROD, and assign it to Q_SHIPMENT in Group Screen One.

Create a New Group

The New Group: G_PROD

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Group Settings								
	Group Name	Query	Print Direction	Matrix Group	Page Break			
^	G_CUST	Q_CUST	Down		Conditional			
	G_PROD	Q_SHIPMENT	Down					
	G_SHIPMENT	Q_SHIPMENT	Down					
	G_SUMMARY	Q_SUMMARY	Down					
v								
Enter a name for this group.								
Report Name: Master_Detail_Summary_Report							<Replace>	

Then, on Group Screen Two, set the spacing as follows. This will put two blank lines between customers, as well as extra spaces between groups. Also, place the product summary group below the shipment data group and add a blank line before the summary group for readability. The master/ detail portion of the master/detail/summary report has now been established with one query and two groups.

Add Group Spacing

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Group Settings								
	Group Name	Relative Position	Lines Before	Spaces Before	Spacing Record Field		Fields Across	
^	G_CUST			2	2			
	G_PROD			2				
	G_SHIPMENT			2				
	G_SUMMARY	Below	1	0				
v								
	Choose the position of this group in relation to the previous group.							
	Report Name: Master_Detail_Summary_Report						<List><Replace>	

Field Settings

To complete the definition of G_PROD, assign the DESCRIP field to it in Field Screen One. A single G_PROD record will now appear each time anew product is found. Notice that you can edit field labels.

Assign a Field to G_PROD

Field in G_PROD

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Field Settings								
	Field Name	_Source	Group	Label				
^	CUSTID	Q_CUST.CUSTID	G_CUST	Custid				
	NAME	NAME	G_CUST	Customer				
	COMMENTS	COMMENTS	G_CUST	Comments				
	DESCRIP	Q_SHIPMENT.DESR	G_PROD	Product				
	CUSTID2	Q_SHIPMENT.CUSTI	G_SHIPMENT	Custid				
	SHIPDATE	SHIPDATE	G_SHIPMENT	Date				
	ITEMTOT	ITEMTOT	G_SHIPMENT	Cost				
	CUSTID3	Q_SUMMARY.CUSTID	G_SUMMARY	Custid				
	DESCRIP2	Q_SUMMARY.DESCRI	G_SUMMARY	Product				
v	TOTAL	TOTAL	G_SUMMARY	Total				
	Enter a name for this field.							
	Report Name: Master_Detail_Summary_Report						<Replace>	

Next, format the fields using the Field Width and Display Format attributes in Field Screen Two

Specify Field Attributes

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help	
Field Settings									
			Field Name	Data Type	Field Width	Display Format	Relative Position	Lines Before	Spaces Before
^			CUSTID	NUM	6				
			NAME	CHAR	20				
			COMMENTS	CHAR	50				
			DESCRIP	CHAR	25				
			CUSTID2	NUM	6				
			SHIPDATE	DATE	9				
			ITEMTOT	NUM	10	ZZ,ZZ9.99			
			CUSTID3	NUM	6				
			DESCRIP2	CHAR	25				
			TOTAL	NUM	10	ZZ,ZZ9.99			
V									
	<							>	
Enter the width of this field in spaces.									
Report Name: Master_Detail_Summary_Report							< Replace >		

In Field Screen Three, suppress printing of the identical fields used to join the queries CUSTID, CUSTID2, and CUSTID3. Also, make the customer name a Wrap field so that longer names will wrap onto the next line. Similarly, make COMMENTS a Wrap field of width 50.

Text Edits

The last step in the Master/Detail/Summary Report is to make a few adjustments in the Text Screens. Delete the column header for the COMMENTS field, and remove the COMMENTS field itself from G_CUST body and place it in the footer of G_SUMMARY instead, so that the remarks will appear after all of the other information about each customer. Then adjust the column headings for G_SUMMARY in the Text Screen.

Edit the Column Heading

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Text Settings								
Object: G_SUMMARY			Type: Column Heading			Status: Edited		
Relative Position: Lines Before: Spaces Before: Width:			Repeat On Page Overflow: X Justification: Left Frequency:					
Text								
Panel Number: 1						Panels Defined: 1		
^ Customer Product Totals _____ v								
Choose the position of this text in relation to the previous text.								
Report Name: Master_Detail_Summary_Report <Normal> <List><Replace>								

This Master/Detail/Summary report has demonstrated how to combine queries to create a sophisticated multi-part report with many levels of detail.

Product Integration

The power of SQL*ReportWriter does not lie solely in its ability to create a variety of different reports. SQL*ReportWriter has a standardized call interface to other Oracle Tools such as SQL*Forms and SQL*Menu, as well as access to your own 3GL-based programs, giving you significant flexibility in creating fully functional application systems. SQL*ReportWriter even gives you the power to send your reports to other users through Oracle*Mail. The tight product integration between SQL*ReportWriter and other Oracle Application Tools will enable you to successfully build and maintain enterprise-wide applications across heterogeneous computing environments.

CHAPTER

5

SUMMARY

Variety of Reports with Precise Control

SQL*ReportWriter, Oracle's premiere reporting tool for application developers, gives you the power to develop a wide range of production-quality reports with complete formatting control.

You can build anything — from a one-minute tabular report, to a sophisticated multi-query report with many subsections.

SQL*ReportWriter enables you to build the reports you need — quickly and easily.

You always retain complete control over every aspect of your report. While you get default settings for many basic layout attributes, you can override any default to create exactly the desired effect.

And SQL*ReportWriter lets you format at many different levels — using multiple SELECT statements, manipulating fields of data, positioning groups of fields, and even fine-tuning with boilerplate text and detailed editing.

Increased Productivity with Non-Procedural Interface SQL*ReportWriter lets you build reports quickly and maintain them easily.

You define your report through the sophisticated fill-in-the-form interface, which includes instant validation checks to catch errors immediately. Your productivity is further enhanced by SQL*ReportWriter's comprehensive online help system – with easy access to context-sensitive help directly from report definition screens, the table of contents, or the index. A List of values is provided for many fields, including a list of available tables and their associated columns for defining queries.

Power of ORACLE's SQL Language

SQL*ReportWriter fully supports the industry standard SQL language for data access. You can specify your report by combining data from multiple databases, including IBM's DB2 and SQL/DS. ORACLE increases the power of SQL even further with more than fifty SQL extensions for calculations, flexible joins, and set operators.

Backed by the power of ORACLE and its family of application development products, SQL*ReportWriter is the reporting tool that's setting the standard for all others.

**BUILDING REPORTS
WITH
SQL*REPORTWRITER[®]**

VERSION 1.1

ORACLE[®]

The Relational Database Management System

Building Reports with SQL*ReportWriter Version 1.1

Part No. 5412-V1.1 0292

Contributing Author: Karen Denchfield-Masterson

Contributors Steve Breyer, Mylan Bui, Mindy Butterfield,
Saleem Haque, Nimish Mehta, Rita Morin,
Jerry Sherman, Dave Simson, Larry Stevens,
Andre Touma, Lisa Van Halsema.

Copyright ©Oracle Corporation 1989,1991

This software/documentation contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this software/documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data - General, including Alternate III (June 1987).

The information 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.

ORACLE, SQL*Forms, SQL*Net, SQL*Plus, SQL*ReportWriter, and SQL*Star are registered trademarks of Oracle Corporation. Oracle*Mail is a trademark of Oracle Corporation.

Ada is a registered trademark of the U.S. Department of Defense. VMS and DEC are registered trademarks of Digital Equipment Corporation. UNIX is a registered trademark of AT&T Bell Laboratories. HP and LaserJet are registered trademarks of Hewlett-Packard Corporation. PostScript is a trademark of Adobe Systems Inc.



PREFACE

Purpose

Building Reports with SQL*ReportWriter contains detailed descriptions of how to create several introductory and many advanced reports with SQL*ReportWriter. It also provides a tutorial and conceptual material to help users get started.

Audience

The information in this book is intended for application developers and report designers. Readers should have a working knowledge of SQL or ORACLE database concepts.

How this Book Is Organized

Building Reports with SQL*ReportWriter is organized into the following parts

Part I: Using SQL*ReportWriter This part contains a tutorial to help you get started using SQL*ReportWriter.

Part II: Understanding SQL*ReportWriter This part introduces SQL*ReportWriter concepts and provides a foundation for using the product.

Part III: Example Reports This part is made up of three chapters:

- Chapter 3 contains a list of all SQL*ReportWriter features found in the example reports, and a sample output of each report with call-outs that display each report's features.
- Chapter 4 contains explanations of how to create 13 introductory reports and is written for new users of SQL*ReportWriter.

- Chapter 5 contains explanations of how to create 23 advanced reports and is written for advanced users of SQL*ReportWriter.
- Chapter 6 contains explanations of how to pass parameters from a SQL*Forms application to a SQL*ReportWriter report, and how to call a report from a 3GL program using the SQL*ReportWriter Call Interface.

How to Use this Book

This book was designed for users with varying levels of experience with SQL*ReportWriter. Below are suggestions on how to use this book, based on your experience: first-time users, experienced users of Version 1.0, and experienced users of Version 1.1.

First Time SQL*ReportWriter Users: You want to learn SQL*ReportWriter quickly so you can start building reports. We suggest the following steps:

1. Go through the entire tutorial in Chapter 1. Take enough time to understand why you enter the words and numbers that you do. (If you ever don't understand why you are entering something, stop and execute the report. When you execute a report, the output "thus far" is displayed. Then, do the next step in the Tutorial, and then execute the report again to see how the report output changed.)
2. Read through Chapter 2, "SQL*ReportWriter Concepts," for a conceptual overview of the product.
3. Read through and build all of the reports in Chapter 4, "Introductory Reports"; then try to build some of your own reports.
4. Skim through the "SQL*ReportWriter Concepts" chapter again, this time learning more about particular features that are discussed.
5. Finally, build any or all of the advanced reports in Chapter 5, "Advanced Reports."

Experienced SQL*ReportWriter Version 1.0 Users: You already know how to create queries and manage groups. You want to know how to build more complex reports that use the new Version 1.1 features. We suggest that you use this book in the following way:

1. Review the introductory report outputs in Chapter 3 (they are labeled as "Level: Introductory" at the end of the report descriptions). If you see an introductory report that you do not know how to build, turn to the detailed explanation of that report in Chapter 4, and build it.

2. Turn to Appendix F, "Version 1.0 to 1.1 Changes," in the SQL*ReportWriter Reference Manual, and read through the changes. Look some of them up in the Manual until you feel comfortable with the new version.
3. Read about and build any or all of the advanced reports found in Chapter 5.

Experienced SQL*ReportWriter Version 1.1 Users: Chapter 5, "Advanced Reports," was created for you. You already know SQL*ReportWriter fairly well. You want to know how to build some advanced reports, and you want to know how to add one or more advanced features to your existing report(s). There are two ways that you might use this book:

- Flip through the report outputs in Chapter 3, find a feature in one of the sample outputs, go to that report's detailed explanation in Chapter 4 or 5, find the feature* in the "Steps to Build this Report" section, and follow the instructions beginning at that point.
- Go through all of the example reports in Chapter 5 so that you can learn techniques that are used to build complex reports.

*In Chapter 4 and 5, features are located in the left margin of the page, and can be identified by a check mark, followed by an italicized feature name (e.g., ✓ Control Break).

Related Publications

Along with this manual, you may want to refer to other documents published by Oracle Corporation:

- SQL*ReportWriter Installation Guide or System Release Bulletin. This document is different for each hardware/software platform. Ask your sales representative for the appropriate part number.
- SQL*ReportWriter Reference Manual, Part No. 641-V1.1
- SQL Language Reference Manual, Part No. 778-V5.1 for use with ORACLE RDBMS Version 5.1
- SQL Language Reference Manual, Part No. 778-V6.0 for use with ORACLE RDBMS Version 6.0 with (or without) the transaction processing option.

Notational Conventions

The following notational conventions are used in this manual:

Font Change [function key]

Enter text exactly as shown.

Represents a SQL*ReportWriter function key. You can refer to the SQL*ReportWriter Keyboard Layout (template) or to the Show Keys screen (online) for a mapping of these function keys to physical keys on your keyboard.

UPPERCASE

Indicates command names, table names, or field names.

Italics

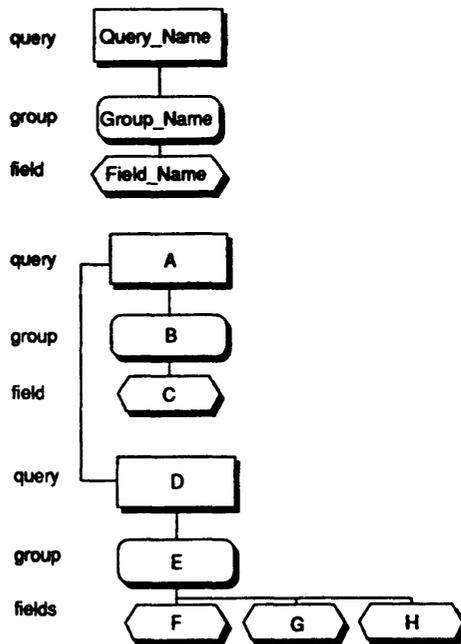
Indicates options for SQL*ReportWriter settings.

✓ feature name

Indicates that steps to build the feature are found beginning at that point.

(VN.N)

Is the Version of SQL*ReportWriter that you will need to build the example report (where N is some number).



The following organization chart conventions are used in this manual:

- queries are indicated by a rectangle with the query name inside
- groups are indicated by a rounded rectangle with the group name inside
- fields are indicated by a polygon with the field name inside.

There are no arrows used when connecting the queries, groups, and fields because the hierarchy is implied. For example, the organization at the left indicates that there is one query (A) that owns one group (B), and that group(B) in turn owns one field (C).

If two queries are joined, that relationship is indicated by a line that connects the parent query to the child query. In the example to the left, D is the child of the A query, and the default group of D(E) owns fields F, G, and H.

New Reports

Several new reports have been added to this book since it was first published:

Type of Report	Description	Page
Conditional Highlighting	Highlights fields when values meet a user-specified criteria	5-10
Ranking	Ranks customers by number, and by a percentage of total sales	5-71
Changeable Number of Records per Column Report	At runtime, formats the user-specified number of records that appear in each column.	5-76
Conditional Printing	Displays specific text and information only when certain criteria are met	5-78

A related concept, the Matrix Ranking Report, has been added to the Matrix Report and can be found on page 4-34.

New Chapter

A new chapter has been added to this book. Chapter 6 describes the concepts and building steps for passing parameters from a SQL*Forms application to a SQL*ReportWriter report, and for calling a report from a 3GL program using the SQL*ReportWriter Call Interface.

Your Comments Are Welcome

We value and appreciate your comments as an ORACLE user. As we write, revise, and evaluate our work, your opinions are the most important input we receive. At the back of this manual is a Reader's Comment Form; we encourage you to use this form to tell us both what you like and dislike about this (or other) Oracle manuals. If the form is gone, or you would like to contact us, please use the following address and phone number:

SQL*ReportWriter Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, California 94065
(415) 506-7000



CONTENTS

PART I	USING SQL*REPORTWRITER	
CHAPTER 1	TUTORIAL	1-1
	Conventions	1 - 2
	Key Functions	1 - 2
	Procedures	1 - 2
	Errors	1 - 3
	SQL*ReportWriter Tables	1 - 3
	The Help System	1 - 3
	Sample Tables	1 - 3
	Lesson I: Creating a Default Report	1 - 5
	Getting Started	1 - 5
	Using the SQL*ReportWriter	1 - 6
	Selecting the Action Menu	1 - 6
	Entering a Query	1 - 7
	Executing Your Report	1 - 8
	Summary of Lesson I	1 - 9
	Lesson II: Creating and Formatting a Break Report	1-10
	Formatting the Report	1-12
	Creating a Summary Field	1-16
	Viewing the Report	1-18
	Summary of Lesson II	1-18

Lesson III: Enhancing the Report	1 - 19
Copying a Report	1 - 20
Entering a Second Query	1 - 21
Positioning and Spacing Groups	1 - 22
Altering Fields	1 - 23
Editing Text	1 - 27
Summary of Lesson III	1 - 28
Lesson IV: Creating a Wrap Field and a Computed Field	1 - 29
Creating a Wrap Field	1 - 30
Changing the Report Width	1 - 31
Creating a Computed Field	1 - 31
Editing Text	1 - 33
Highlighting Text	1 - 34
Summary of Lesson IV	1 - 36

PART II

UNDERSTANDING SQL*REPORTWRITER

CHAPTER 2

SQL*REPORTWRITER CONCEPTS	2 - 1
Elements of a Report	2 - 2
Report Building Process	2 - 3
Creating a Blank Report	2 - 3
Specifying the Data	2 - 3
Specifying the Report Format	2 - 3
SQL*ReportWriter Objects	2 - 4
Query Objects	2 - 5
Multiple Queries	2 - 5
Field Objects	2 - 10
Field Settings	2 - 10
Group Objects	2 - 13
Managing Groups	2 - 13
Group Settings	2 - 14

Summary Objects	2-18
Print Group	2-19
Reset Group	2-19
Summary Functions	2-20
Text Objects	2-22
Report and Page Text Objects	2-22
The Report Object	2-28
Page Layout	2-28
Parameter Form Defaults	2-29
Security	2-29
History	2-29
Parameter Object	2-29
Run-time Parameter Form	2-30
Element of SQL*ReportWriter	2-30
SQLREP	2-30
RUNREP	2-30
DUMPREP/LOADREP	2-30
GENREP	2-31
PRINTDEF	2-31
TERMDEF	2-31
MOVEREP	2-31

PART III

BUILDING REPORTS

CHAPTER 3

REPORT FEATURES AND OUTPUTS	3-1
Features Listing	3-2
Report Name Listing	3-4
Visual Index	3-6
Tabular Report	3-6
Subtotal Report (V1.0)	3-7
Computed Fields Report (V1.0)	3-8
Computations On Summaries: using cross-referenced queries (V1.1)	3-9
Computations On Summaries: using SQL (V1.0)	3-10

Time Series Calculations Report (V1.0)	3-11
Ranking Report (V1.1)	3-12
Break Report (V1.0)	3-13
Wrapped Break Report (V1.1)	3-14
Intermixing Fields From Different Groups using the Margin setting (V1.1)	3-15
Aggregating Data Within Ranges Report (V1.0)	3-16
Across Reports With Control Breaks (V1.0)	3-17
Master/Master Report (V1.0)	3-18
Master/Detail Report (V1.0)	3-19
Suppressing Column Headers When No Detail Records Are Retrieved (V1.0)	3-20
Master/Detail/Summary Report (V1.0)	3-21
Matrix Report (V1.0)	3-22
Matrix Break Report (V1.0)	3-23
Matrix With Zeros For Null Values (V1.0)	3-24
Page Heading Report (V1.0)	3-25
Placing Database Values In Page Headings (V1.1)	3-26
Group Footing Report (V1.0)	3-27
Relative Positioning Of Unrelated Groups: using null queries (V1.0)	3-28
Relative Positioning Of Related Groups: using null queries (V1.0)	3-29
Creating Groups With No Printable Fields (V1.1)	3-30
Reports With A Changeable Number Of Records Per Column (V1.1)	3-31
Conditional Printing (V1.1)	3-32
Conditional Highlighting (V1.1)	3-33
Introductory Mailing Label Report (V1.0)	3-34
Advanced Mailing Label Report (V1.1)	3-35
Form Letter Report (two techniques one with V1.0, one with (V1.1)	3-36
Check Printing Report (V1.1)	3-37
Spelling Out Cash Amounts On Checks (V1.1)	3-38
Printing Reports On Pre-printed Forms (V1.1)	3-39
Invoice Report (V1.1)	3-40

CHAPTER 4	INTRODUCTORY REPORTS	4-1
	How To Use This Chapter	4-2
	How To Load Introductory Reports	4-2
	Tabular Report (V1.0)	4-3
	Distinguishing Features	4-3
	Concepts	4-3
	Organization	4-3
	Steps to Build this Report	4-4
	Break Report (V1.0)	4-6
	Distinguishing Features	4-6
	Concepts	4-7
	Organization	4-7
	Steps to Build this Report	4-8
	Related Concepts	4-9
	Subtotal Report (V1.0)	4-11
	Distinguishing Features	4-11
	Concepts	4-11
	Organization	4-13
	Steps to Build this Report	4-13
	Introductory Mailing Label Report (V1.0)	4-16
	Distinguishing Features	4-16
	Concepts	4-16
	Organization	4-17
	Steps to Build this Report	4-17
	Computed Field Report (V1.0)	4-20
	Distinguishing Features	4-20
	Concepts	4-21
	Organization	4-21
	Steps to Build this Report	4-21

Form Letter Report (two methods: one with V1.0, one with (V1.1))	4-24
Distinguishing Features	4-24
Concepts	4-25
Organization	4-25
Steps to Build this Report	4-25
Matrix Report (V1.0)	4-30
Distinguishing Features	4-30
Concepts	4-30
Organization	4-31
Steps to Build this Report	4-32
Related Concepts (V1.1)	4-34
Master/Master Report (V1.0)	4-35
Distinguishing Features	4-35
Concepts	4-35
Organization	4-35
Steps to Build this Report	4-36
Master/Detail Report (V1.0)	4-38
Distinguishing Features	4-38
Concepts	4-38
Organization	4-39
Steps to Build this Report	4-39
Master/Detail/Summary Report (V1.0)	4-41
Distinguishing Features	4-41
Concepts	4-41
Organization	4-42
Steps to Build this Report	4-43
Group Footing Report (V1.0)	4-46
Distinguishing Features	4-46
Concepts	4-46
Organization	4-47
Steps to Build this Report	4-47

	Page Heading Report (V1.0)	4-49
	Distinguishing Features	4-49
	Concepts	4-49
	Organization	4-50
	Steps to Build this Report	4-50
CHAPTERS		
	ADVANCED REPORTS	5-1
	How To Load Advanced Reports	5-2
	Intermixing Fields From Different Groups: using the Margin setting (V1.1)	5-3
	Distinguishing Feature	5-3
	Concepts	5-3
	Organization	5-4
	Steps to Build this Report	5-4
	Creating Groups With No Printable Fields (V1.1)	5-6
	Distinguishing Features	5-6
	Concepts	5-7
	Organization	5-7
	Steps to Build this Report	5-7
	Conditional Highlighting (V1.1)	5-10
	Distinguishing Features	5-10
	Concepts	5-10
	Organization	5-11
	Steps to Build this Report	5-11
	Relative Positioning Of Unrelated Groups: using null queries (V1.0)	5-14
	Distinguishing Features	5-14
	Concepts	5-15
	Organization	5-16
	Steps to Build this Report	5-16

Relative Positioning Of Related Group: using null queries (V1.0)	5-18
Distinguishing Features	5-18
Concepts	5-18
Organiation	5-20
Steps to Build this Report	5-20
Suppressing Column Headings When No Detail Records Are Retrieved (VM)	5-22
Distinguishing Features	5-22
concepts	5-22
Organization	5-23
Steps to Build this Report	5-23
Placing Database Values In Page Headings (V1.1)	5-25
Distinguishing Features	5-25
Concepts	5-25
Organization	5-26
Steps to Build this Report	5-26
Aggregating Data within Ranges (V1.0)	5-28
Distinguishing Features	5-28
Concepts	5-28
Organization	5-29
	5-29
Printing Reports On Pm-printed Forms (V1.1)	5-30
Distinguishing Features	5-30
Concepts	5-30
Organization	5-32
Steps to Build this Report	5-32
Across Reports With Control Breaks (V1.0)	5-35
Distinguishing Features	5-35
Concepts	5-35
Organization	5-36
Steps to Build this Report	5-36

Wrapped Break Report (V1.1)	5-38
Distinguishing Features	5-38
Concepts	5-38
Organization	5-39
Steps to Build this Report	5-39
Computation On Summaries: using cross-referenced queries (V1.1)	5-40
Distinguishing Features	5-40
Concepts	5-40
Organization	5-41
Steps to Build this Report	5-42
Computation On Summaries: using SQL (V1.1)	5-44
Distinguishing Feature	5-44
Concepts	5-44
Organization	5-45
Steps to Build this Report	5-45
Time Series Calculations (V1.0)	5-48
Distinguishing Features	5-48
Concepts	5-48
Organization	5-49
Steps to Build this Report	5-49
Advanced Mailing Label Report (V1.1)	5-50
Distinguishing Features	5-50
Concepts	5-50
Organization	5-51
Steps to Build this Report	5-51
Invoice Report (V1.1)	5-52
Distinguishing Feature	5-52
Concepts	5-53
Organization	5-54
Steps to Build this Report	5-54

Check Printing Report (V1.1)	5-57
Distinguishing Features	5-57
Concepts	5-57
Organization	5-58
Steps to Build this Report	5-58
Spelling Out Cash Amounts On Checks (V1.1)	5-60
Distinguishing Features	5-60
Concepts	5-60
Organization	5-61
Steps to Build this Report	5-61
Matrix With Zeros for Null Values (V1.0)	5-63
Distinguishing Features	5-63
Concepts	5-63
Organization	5-64
Steps to Build this Report	5-64
Matrix Break Report (V1.0)	5-66
Distinguishing Features	5-66
Concepts	5-66
Organization	5-47
Steps to Build this Report	5-67
Ranking Report (V1.1)	5-71
Distinguishing Features	5-71
Concepts	5-71
Organization	5-72
Steps to Build this Report	5-72
Related Concepts	5-74
Steps to Build this Report	5-74
Changeable Number Of Records per Column Report (V1.1)	5-76
Distinguishing Features	5-76
Concepts	5-76
Organization	5-76
Steps to Build this Report	5-77

	Conditional Printing (V1.1)	5-78
	Distinguishing Features	5-78
	Concepts	5-78
	Organization	5-79
	Steps to Build this Report	5-79
CHAPTER 6	INTEGRATING SQL*REPORTWRITER	6-1
	Integrating SQL*ReportWriter and SQL*Forms	6-2
	Concepts	6-2
	Steps to Build this Report	6-2
	Trigger Syntax Using PL/SQL-based Triggers	6-3
	Trigger Syntax without PL/SQL Triggers	6-3
	Integrating a Form and Report for Summit Sporting Goods	6-4
	Calling SQL*ReportWriter from a 3GL Program	6-5
	Examples	6-5
GLOSSARY		
INDEX		

PART

I



**USING
SQL*REPORTWRITER**

1

TUTORIAL

This tutorial, presented in four lessons, introduces you to SQL*ReportWriter. The tutorial works you through some basic tasks that you'll often do when creating your own reports. At the end of each lesson you will have a completed report and can leave the Tutorial; however, go through all of the lessons if you can.

Lesson I teaches you how to create a default tabular report. You will learn how to:

- invoke SQL*ReportWriter
- create a report
- enter a query
- execute the report
- view your report output.

In Lesson II you will modify the report to create a break, or outline-style, report. You will also make some easy formatting changes which will introduce you to the various parts of a report definition. You will learn how to:

- create a control break
- edit column headings
- modify spacing
- add a page header
- compute the total of a report field.

In Lesson III you will create and modify a master/detail report. You will learn how to:

- create a copy of your first report
- add a second query
- link the two queries into a master/detail relationship
- edit field definitions
- edit text
- use all of the report browsing facilities.

Finally, in Lesson IV you will enhance your report, adding two different types of fields to it. You will learn how to:

- modify a query
- cause a field to word-wrap
- change the width of your report
- edit text using the query mode
- highlight text.

Conventions

The following conventions are used throughout this book.

Key Functions

SQL*ReportWriter function names are used instead of key names. For example, “press [Select]” means press the key that invokes the Select function; “press [Next Field]” means press the key that moves the cursor to the next field. Use the help-system to find specific key mappings. Alternatively, refer to your ORACLE Installation and User's Guide or System Release Bulletin for these mappings.

On most devices the Return key is context-sensitive and is mapped to either [Next Field] or [Select]. In multi-line fields, the Return key inserts a new line.

Procedures

Some procedures are not repeated throughout the entire tutorial. For example, the procedure “press [Select] to select the Action Menu” is shortened to “select the Action Menu.” Likewise, “press [Next Field repeatedly until the cursor is in the SELECT Statement area” changes to “move to the SELECT Statement area.”

Errors

If you make a mistake that causes SQL*ReportWriter to generate an error message, press the Return key to clear the error message before you correct the problem. Remember, any time you receive an error message you can press [Help] while the message is still on the screen. A help screen will appear explaining why you got the error and what you can do to fix it.

If you have made a SQL error, an asterisk will appear in the error message at the point at which the error occurred. SQL error messages are several lines long, so you may need to press [Next Line] several times to see the exact line in your SQL Statement where the error occurred.

If you should make an error and wish to discard your changes to the current screen, press [Undo]. **Note:** On some keyboards the Undo function is mapped to pressing the same key twice (e.g., pressing F20 two times on a VT220 executes the Undo function).

SQL*ReportWriter Tables

In order to use SQL*ReportWriter, several tables must be installed on your ORACLE database. See your installation guide for details. The tables are described in Appendix C of the SQL*ReportWriter Reference Manual.

The Help System

SQL*ReportWriter's on-line help system provides context-sensitive information about report settings, as well as general topic information. For example, if you are in a report definition screen and need information about the current setting, press [Help] and a screen of information about that setting appears. If there is more than one screen for the current setting, press [Scroll Down]. When you are finished reading the help, select Quit from the menu (press [Next Field] and [Previous Field] until you reach Quit, and then press [Select]) to return to where you were when you pressed [Help]. You are encouraged to use the help system throughout this tutorial.

Sample Tables

While you are using the tutorial, you will use data from a sporting goods manufacturer named "Summit Sporting Goods." Summit has created three basic tables to record orders for products, available items, and customer information.

The Ord table contains information about the orders that are placed by customers, the Item table contains information about items in an order, and the Customer table contains information about each customer. The contents of these tables is defined as follows

The Ord Table

Table Column	Description
ORDID	order identification number
ORDERDATE	date that the order was issued
SHIPDATE	date that the order was shipped
TOTAL	total cost of all the items in the order
CUSTID	customer identification number
COMMPLAN	commission plan used

The Item Table

Table Column	Description
ORDID	identification number of the associated order
PRODID	product identification number
QTY	quantity of the item ordered
ITEMTOT	total cost of the item in the order
ACTUALPRICE	unit price of the item
ITEMID	identification number of the item

The Customer Table

Table Column	Description
CUSTID	customer identification number
NAME	name of the customer
ADDRESS	address of the customer
CITY	city of the customer
STATE	state of the customer
ZIP	zip of the customer
AREA	area/region to which customer is assigned
PHONE	phone number of the customer
REPID	identification number of the sales representative
CREDITLIMIT	maximum credit the customer is allotted
COMMENTS	comments about the customer

These tables are part of the demo database that is shipped with SQL*ReportWriter. Should you have any trouble accessing the tables, contact your Database Administrator (DBA).

Lesson I: Creating a Default Report

Lesson I of this tutorial teaches you how to produce a tabular report using most of the default settings. The report you will produce is shown in Figure 1-1.

FIGURE 1-1
Report Output

Ordid	Shipdate	Custid	Total
620	12-MAR-87	100	4450
621	01-JAN-87	100	730
610	08-JAN-87	101	101.4
611	11-JAN-87	102	45
610	05-MAR-87	102	3555.5
614	05-FEB-87	102	23040
616	10-FEB-87	103	754
612	20-JAN-87	104	5310
613	04-FEB-87	104	300
617	03-MAR-87	105	46370
615	05-FEB-87	107	710
613	01-FEB-87	108	10760

Getting Started

This section describes how to invoke SQL*ReportWriter. To use SQL*ReportWriter, you will need a username and password for your ORACLE database. Most likely, they will be different from the username and password you used to log onto your computer. Contact your Database or System Administrator for details.

1. Logon to your computer using normal procedures.
2. At the operating system prompt, type the command:

```
sqlrep username /password
```

and then press Return.

(If you get a "notables" error message, contact your Database Administrator.)

Using the SQL*ReportWriter Screens

Each choice on the Main Menu line (except Action) accesses a different set of SQL*ReportWriter screens. The Help choice accesses the help system. You navigate to a SQL*ReportWriter screen by selecting the corresponding menu choice. Select a menu choice by placing the cursor on the choice you want and pressing [Select]. (On some operating systems, [Select] is mapped to the Return key.) You can also press the first letter of the choice you want; this positions the cursor and selects the menu simultaneously.

Selecting the Action Menu

In order to create a new report, you must select the Action menu and then select the New option. By default, the Action choice is highlighted when you enter the Main Menu.

1. Press [Select] to choose the Action menu (or press A).
2. Press [select] to select New.
3. Press [Help] to learn what naming conventions your report name must follow.
4. By default the Quit option on the Help menu line is highlighted. Press [Select] to exit the help system and return to the screen you left when you entered the help system.
5. For the purpose of this tutorial, enter `FirstReport`. See Figure 1-2.

FIGURE 1-2
Creating a New Report

The screenshot shows the main menu of SQL*ReportWriter. The menu items are: Action, Query, Group, Field, Summary, Text, Report, Parameter, and Help. The 'Action' menu is highlighted. Under the 'Action' menu, the 'New' option is selected. Below the 'New' option, there are several sub-options: Open, Copy, Rename, Drop, Execute, Generate, and Quit. The 'Rename' option is active, and the 'Name of New Report:' field is visible. The field contains the text 'FirstReport'. At the bottom of the screen, there is a prompt: 'Enter a name for the new report.' and a 'Report Name:' label. The prompt is highlighted, and the text '<List><Replace>' is visible at the bottom right.

6. Press [Accept].

Entering a Query

After you name the new report you must specify the data that will be used in the report. The Query Screen lets you select data using SQL statements. You must first name a query and then enter a SQL statement.

Select Query from the Main Menu.

Enter `ORDTABLE` in the Query Name field. Every query must be named so that you can reference it elsewhere in the report definition. We decided to call this query `Ordtable` because the `SELECT` statement queries information from the `Ord` table. You can, however, use any naming convention for your queries: you can append numbers to them, etc. Press [Help] if you want to learn more about query naming restrictions.

Press [Next Field]; the cursor moves to the SQL Statement entry area.

Enter the following SQL statement. While entering the statement, press the Return key to move your cursor to the next line (see Figure 1-3).

```
SELECT ORDDID, SHIPDATE, CUSTID, TOTAL
FROM ORD WHERE SHIPDATE>' 31-DEC-86'
ORDER BY CUSTID
```

FIGURE 1-3
Query Screen

The screenshot shows a software interface for defining a query. At the top, there is a menu bar with options: Action, Query, Group, Field, Summary, Text, Report, Parameter, Help. Below the menu bar is a section titled "Query Settings". Inside this section, there are two main input areas: "Query Name: ORDTABLE" and "SELECT Statement". The "SELECT Statement" area contains the following text: "SELECT ORDDID, SHIPDATE, CUSTID, TOTAL FROM ORD WHERE SHIPDATE>'31-DEC-86" ORDER BY CUSTID". Below the "SELECT Statement" area, there are two fields for "Parent-ChildRelationships": "Parent Query 1:" and "Parent Query 2:". Below these fields is a table with three columns: "Child Columns", "Parent 1 Columns", and "Parent 2 Columns". At the bottom of the screen, there is a prompt: "Enter a SQL SELECT Statement defining data for this report" and a field for "Report Name: FirstReport" with a "<Replace>" button.

This manual assumes you are familiar with SQL. If you need a quick review, refer to the on-line help system chapter, "SQL."

5. Press [Accept] to return to the Main Menu.

Executing Your Report

At this point you can execute FirstReport. The Action Menu contains the option for executing a report. You can also execute a report from the operating system command line using RUNREP. See the "Run-time Interface" in Chapter 4 of the SQL*ReportWriter Reference Manual for more information. In this tutorial you will run your report from the Action Menu.

1. Choose the Action menu and select Execute. By default, FirstReport appears in the dialog box because it is the open report.
2. Press [Accept] to acknowledge that you want to execute the report named FirstReport.

When you execute the report, the Run-time Parameter Form appears, prompting you for information about print options and parameters.

Run-time Parameter Form

Each time you run a report in interactive mode, the Run-time Parameter Form (Figure 1-4) pops up before the report output is displayed. This form allows you to enter run-time parameters.

FIGURE 1-4
Run-time Parameter Form

Parameter	Value
Destination Type	Screen
File Name/Spool Device	FirstReport.lis
Printer Description File	dflt
Number of Copies	1

Enter the desired value for each parameter.

Report Name: FirstReport <Replace>

3. The defaults will do for now, so just press [Accept] when this form appears. For more information about the Run-time Parameter Form, refer to Chapter 4 in the SQL*ReportWriter Reference Manual.

Since you used the default settings for device type and device name on the Report Screen, your screen output should look like the report in figure 1-5.

FIGURE 1-5
Viewing Report Output

<u>Ordid</u>	<u>Shipdate</u>	<u>Custid</u>	<u>Total</u>
620	12-MAR-87	100	4450
621	01-JAN-87	100	730
610	08-JAN-87	101	101.4
611	11-JAN-87	102	45
618	05-MAR-87	102	3555.5
614	05-FEB-87	102	239.00
616	10-FEB-87	103	764
612	20-JAN-87	104	5310
619	04-FEB-87	104	300
617	03-MAR-87	105	46370
615	05-FEB-87	107	710
613	01-FEB-87	100	10760

4. Press [Accept] to return to the Main Menu.

Summary of Lesson I

In Lesson I of this tutorial, you learned how to create and run a default report. This included the following steps:

- invoking SQL*ReportWriter with the SQLREP command
- creating the report by selecting New from the Action menu
- defining the query by entering a SQL statement on the Query Screen

- running the report by selecting Execute from the Action menu.

You can now leave the Tutorial with a completed report. The next lesson, however, teaches you how to build break reports a type of report that is often built.

Lesson II Creating and Formatting a Break Report

In Lesson II of this tutorial, you will learn how to create a break report, format it, add a summary field, and then view the report output in two different modes. A break report contains several detail records for every master record retrieved. To create a break report, you must do the following

- create a new group
- associate a query to the new group
- position the groups
- assign at least one field to the new group.

One group is created by default for each query, and that group owns all of the columns that are SELECTed in the query. The purpose of groups is to enable you to simultaneously move all columns (or fields) owned by a group to another part of the report, rather than having to move each field individually. At the end of this lesson, your report will look like Figure 1-6.

FIGURE 1-6
Viewing Report Output

```

- ORDER SUMMARY -

Customer Id:    100
Order Id  Shipdate    Total
-----
   620  12-198-87    4450
   621  01-JAN-87     730
Sum                               5180.00

Customer Id:    101
Order Id  Shipdate    Total
-----
   610  00-JAN-87     101.4
Sum                               101.40

```

Create a New Group

1. select Group from the Main Menu by entering the letter g. A group is a distinct set of data. When you query data from the database, the resulting data of that query become one group. SQL*ReportWriter names the default groups by prefixing G to the query name. This report has one group thus far, G_Ordtable, created by default for the data from the Ordtable query.

2. Press [Insert Record Above] to insert a group above G_Ordtable. Note that groups appear in the report output in the same order in which they appear on this screen.
3. Enter G_CUSTOMER in the Group Name column. Note that the G_ that you prefixed is not required. You have now created a new group. Press [Next Held].

Associate a Query to the New Group

4. Enter ORDTABLE in the Query column for G_Customer. This tells SQL*ReportWriter that any fields that are assigned to this group are from the Ordtable query. You have now associated a query with the group.

Position the Groups

5. Press [Scroll Right] once, and then [Next Record] once, so that your cursor is in the Relative Position column of the G_Ordtable group.
6. Enter BELOW to cause the fields assigned to the G_Ordtable group to be located below the fields assigned to the G_Customer group.
7. Press [Prev Record] once and then [Next Field] three times to reach the Record Spacing entry field. Enter a 1. This places one blank line between each customer record.
8. Press [Scroll Right] once to reach Group Screen Three, and then press [Next Field] once so that your cursor is in the Label Position entry field for the G_Customer group.
9. Enter Left. This will cause the CUSTID field label to be placed to the left of the CUSTID field value.
10. Press [Accept] to return to the Main Menu.

Assign at Least One Field to the New Group

11. You will assign a field to the new group because each group must own at least one field. You will assign the CUSTID field because you want one CUSTID record to print for several related records (Orderids, Shipdates, and Totals). Select Field from the Main Menu.
12. Move your cursor to the Group entry field for the CUSTID field by pressing [Next Record] twice, and [Next Field] two times.
13. Press [Delete Word] and enter G_CUSTOMER. Your screen should look like Figure 1-7.

FIGURE 1-7
Assigning New Group

Action Query Group Field Summary Text Report Parameter Help				
Field Settings				
	Field Name	Source	Group	Label
^	ORDID	ORDID	G_ORDTABLE	Ordid
	SHIPDATE	SHIPDATE	G_ORDTABLE	Shipdate
	CUSTID	CUSTID	G_CUSTOMER	Custid
	TOTAL	TOTAL	G_ORDTABLE	Total
V				
				>

Enter the name of the group that contains this field.

Report Name: FirstReport <List><Replace>

14. Execute the report. It should now look like Figure 1-8.

FIGURE 1-8
Viewing Report Output

Custid	100		
Ordid	Shipdate		Total
---	---	---	---
620	12-MAR-87		4450
621	01-JAN-87		730
Custid	101		
Ordid	Shipdate		Total
---	---	---	---
610	08-JAN-87		101.4
Custid	102		
Ordid	Shipment		Total
---	---	---	---
611	11-JAN-87		45
610	06-MAR-87		3555.5
614	05-FEB-87		23940
Custid	103		
Ordid	Shipdate		Total
---	---	---	---
616	10-FEB-87		764

12. Press [Accept] to return to the Main Menu.

Formatting the Report

You can make the information in this report more readable by changing the field labels, modifying the spacing and adding a page header to the report.

Editing and Positioning Field Labels

The labels appear as column headings in your report. By default, the field label is taken from the name of the column or the text of the expression that was queried from the database. You will use the Field Screens and modify the field labels.

1. Select Field from the Main Menu.
2. Move your cursor to the Label entry field for CUSTID, and use the cursor keys and [Delete Backward] to clear the field (or press [Delete Word]). You can also use [Insert/Replace], and type over the existing label.
3. Type in the new label, customer Id:.
4. Move the cursor to the Label entry field for ORDID. Change Ordid to Order Id. your screen will now look like Figure 1-9.

FIGURE 1-9
Field Screen One

Field Settings			
Field Name	Source	Group	Label
CUSTID	CUSTID	G_CUSTOMER	Customer Id:
ORDID	ORDID	G_ORDTABLE	Order Id
SHIPDATE	SHIPDATE	G_ORDTABLE	Shipdate
TOTAL	TOTAL	G_ORDTABLE	Total

Enter the field label for this field.

Report Name: FirstReport <List><Replace>

5. Return to the Main Menu and select Group.
6. Press [Scroll Right] once to go to Group Screen Two.

Adding a Page Header

You will use the Text Screen to add the page header-ORDER SUMMARY-to the report you just created.

10. Select Text from the Main Menu. Your cursor will now be in the text object that has an Object of PAGE and a Type of Header. (This information is found at the top of the Text Settings screen.)
11. Move the cursor to the Justification entry field by pressing [Next Field] several times.
12. Press [List]. This will display a List of values from which to choose.

In the List of values, move the cursor to Center and press [Select]. Center justification causes each line of text entered in the Text entry area to be centered in your report.

Press [Next Field] to move to the Text entry field and type

- ORDER SUMMARY -

Your screen should look like Figure 1-11.

FIGURE 1-11
Creating a Page Header

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Text Settings								
Object: PAGE			Type: Header			Status: Edited		
Relative Position: Lines Before: Spaces Before: Width:			Repeat On Page Overflow: Justification: Center Frequency:					
Panel Number: 1						Panels Defined: 1		
· - ORDER SUMMARY -								
v								
Edit the text.								
Report Name: FirstReport				<Normal>			<Replace>	

15. Press [Accept].

Creating a Summary Field

In this section, you will learn how to create a summary of a field from your query. The field that will be summarized is TOTAL, the total sales of each customer order. You define summaries on the Summary screens.

1. Select Summary from the Main Menu.
2. Enter `Customer_Total` in the Summary Name entry field
3. Press [Next Field] to move to the Field column.
4. Press [List] to see a list of valid fields to summarize (Figure 1-12).

FIGURE 1-12
Using the List of Values

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Summary Settings								
	Summary Name		Field	Function	Data Type	Fields		
^	CUSTOMER_TOTAL					CUSTID ORDID SHIPDATE TOTAL		
V								
Choose a field to summarize.								
Report Name: FirstReport							<Replace>	

- 5.. In the List of values, move the cursor to TOTAL and press [Select]

Notice that Data Type and Width now display default values derived from the definition of TOTAL.
6. Type `SUM` (or select it from the List of values) in the Function column.

This tells SQL*ReportWriter the type of summary function to perform

- While you are on this screen, enter a display format for the summary by pressing [Next Field] until you reach the Display Format entry field and type ZZZZ9.99. (For an explanation of Display Formats, see Chapter 8 in the SQL*ReportWriter Reference Manual.) See Figure 1-13. Note that the formats for summary fields can be different from the format of the data being summarized.

FIGURE 1-13
Summary Screen One

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Summary Settings								
	Summary Name	Field	Function	Data Type	Width	Display Format		
^	CUSTOMER_TOTAL	TOTAL	Sum	NUM	8	ZZZZ.99		
v								
Enter the format mask (e.g. \$999.9)							>	
Report Name: FirstReport							<Replace>	

- Press [Accept] and then execute your report to see the results. Your report should now look like Figure 1-14.

FIGURE 1-14
Report Output

```

                                -ORDER SUMMARY-

Customer Id:      100
  Order_Id  Shipdate      Total
   620  12-MAR-87      4450
   621  01-JAN-87         730
   Sum                               5188.00

Customer Id:      101
  Order_Id:  Shipdate      Total
   610  08-JAN-87      101.4
   Sun                               101.4

```

Viewing the Report

There are two ways to view your report output on the screen: in Browse Mode and in Window Mode.

Browse Mode

When output is displayed on a screen, SQL*ReportWriter is automatically in Browse Mode. This means that you can page up and down through the report output. For example, the next page of the report will show information for Customer Id: 105.

1. Press [scroll Right].
2. Press [scroll Left].

Window Mode

Window Mode allows you to scroll within one page line by line and character by character.

1. Press [Window].
2. Use [Previous Line] and [Next Line] to scroll through the report output.
3. Press [Accept] to exit Window Mode; press [Accept] again to return to the Main Menu.

Summary of Lesson II

In Lesson II of this tutorial you created a break report and made some formatting changes. By doing so, you learned how to do the following

- make a control break by creating a new group and assigning a field to it
- edit field labels
- modify group spacing to change your report format
- create a page header by adding text to a Text Screen
- create a field on the Summary Screen and assign a function to it
- view the report output using the interactive browser (in Browse and Window Modes).

You can now leave the Tutorial with a completed report. The next lesson, however, teaches you how to create a two-query (master/detail) report, and how to create summaries of data selected in queries.

Lesson III: Enhancing the Report

In Lesson III you will build on the report you created in Lesson II, by creating and modifying a two-query report. Figure 1-15 shows the report that you will build in this lesson. It shows specific information about the items purchased in each customer order.

FIGURE 1-15
Multi-Query Report

- ORDER SUMMARY -					
Customer Id: 100					
	Itemid	ProdId	Actualprice	Qty	ItemTot
Order Id: 620	1	100050	35.00	10	350.00
Shipdate: 12-MAR-87	2	200376	2.40	1000	2400.00
Total: 4450.00	3	102130	3.40	500	1700.00
Order Id: 621	1	100051	45.00	10	450.00
Shipdate: 01-JAN-87	2	100070	2.00	100	200.00
Total: 730.00					
Sum			5180.00		
Customer Id: 101					
	Itemid	ProdId	Actualprice	Qty	ItemTot

Copying a Report

First, copy the report you created in Lesson II and name it TwoQueryReport.

1. Select Action from the Main Menu.
2. Select Copy from the Action Menu and then press [Next Field] to move to the Copy To field.
3. Type
TwoQueryReport

Your screen should look like Figure 1-16.

FIGURE 1-16
Copying I Report

The screenshot shows the SQL*ReportWriter application window. The menu bar includes Action, Query, Group, Field, Summary, Text, Report, Parameter, and Help. The 'Action' menu is open, showing options: New, Open, Copy (highlighted), Rename, Drop, Execute, Generate, and Quit. To the right of the menu, the 'Report to Copy From:' field contains 'FirstReport'. Below it, the 'Report to Copy To:' field contains 'TwoQueryReport'. At the bottom of the window, a prompt reads 'Enter a name for the new (copied) report.' Below this prompt, the text 'Report Name: FirstReport' is displayed, followed by '<List><Replace>'.

4. Press [Accept].

Entering a Second Query

In this section you add information about the items in each customer's order. To do this you add a second query that selects data from the Item table. You then create a relationship between the two queries so the items and orders correspond. You will make the query that selects the items subordinate to the Ordtable query. In other words, you are establishing a parent/child or master/detail relationship between the two queries: for each customer order, its associated items are retrieved. Refer to Figure 1-17 as you follow the next procedure.

1. Select Query from the Main Menu.
2. Press [Insert Record Below] to insert a new query.

Notice the query indicator now says Query 2 of 2.

3. Enter `ITEMTABLE` as the Query Name.
4. Type in the following SQL statement

```
SELECT * FROM ITEM  
ORDER BY ITEMID
```

5. Use [Next Field] to leave the SELECT statement. To exit multi-line fields, you must use the explicit Next Field key (i.e., the key that functions only as [Next Field]).

Enter `ORDTABLE` as Parent Query 1, or press [List] and select `ORDTABLE` from the List of values Designating the query from the Ord table as the Parent Query causes the child query, Itemtable, to be executed once for each customer order from the Ord table.

Press [Next Field] until the cursor is in the Child Columns field. Now, using the List of values, select `ORDID` for both the Child Columns and Parent 1 Columns entry areas. This causes information from the Item table to be retrieved only when the order ID number is equal to that of the order ID number from the Ord table.

FIGURE 1-17
Enter a Second Query

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Query Settings								
Query Name: ITEMTABLE						Query 2 of 2		
SELECT Statement								
^	SELECT * FROM ITEM							
	ORDER BY ITEMID							
v								
Parent-Child Relationships								
Parent Query 1: ORDTABLE				Parent Query 2:				
	Child Columns		Parent 1 Column		Parent 2 Column			
^	ORDID		ORDID					
v								
Choose a column to link with the column from the current query.								
Report Name: TwoQueryReport							<Replace>	

8. Return to the Main Menu.

Positioning and Spacing Groups

In this section you position the data from the query you just created.

1. Select Group from the Main Menu.
2. Press [Scroll Right] to go to Group Screen Two.
3. Enter a 1 in the Spacing (Record) entry field for G_Ordtable. This places one blank line between each customer order. Your screen should now look like Figure 1-18.

FIGURE 1-18
Relative Positioning and Spacing

Action Query Group Field Summary Text Report Parameter Help							
Group Settings							
	Group Name	Relative Position	Lines Before	Spaces Before	Spacing Record	Field	Fields Across
^	G_CUSTOMER G_ORDTABLE G_ITEMTABLE	Below	2 1		3 1		
V							

Enter the number of lines to leave blank between each record in this group.

Report Name: TwoQueryReport <Replace>

4. Go to Group Screen Three.
5. Set the Label Position for G_Ordtable to Left. This changes the default setting so that instead of appearing above as column headings, the labels will appear to the left of each field value.
6. Return to the Main Menu.

Altering Fields

In this section you will alter some default labels and display formats, define some fields to be skipped from the report output, and then alter a text format.

1. Select Field from the Main Menu.

2. Move to the second Label entry field, Order Id.
3. Press [Next Word] until you are at the end of the label and enter a colon. Continue to do this for all fields selected from the Ord table (they all have G_Ordtable as their Group) until your screen looks like Figure 1-19.

FIGURE 1-19
Altering the Label
Entry Field

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Field Settings								
			Field Name	Source	Group	Label		
			CUSTID	ORDTABLE.CUSTID	G_CUSTOMER	Customer Id:		
			ORDID	ORDTABLE.ORDID	G_ORDTABLE	Order Id:		
			SHIPDATE	SHIPDATE	G_ORDTABLE	Shipdate:		
			TOTAL	T e x t	G_ORDTABLE	Total:		
			ORDID2	ITEMTABLE.ORDID	G_ITEMTABLE	Ordid		
			ITEMID	ITEMID	G_ITEMTABLE	Itemid		
			PRODID	PRODID	G_ITEMTABLE	Prodid		
			ACTUALPRICE	ACTUALPRICE	G_ITEMTABLE	Actualprice		
			QTY	QTY	G_ITEMTABLE	Qty		
			ITEMTOT	ITEMTOT	G_ITEMTABLE	Itemtot		
v _____ >								
Enter the field label for this field.								
Report Name: TwoQueryReport							<Replace>	

4. Press [Scroll Right] to go to Field Screen Two.
5. Move to the Display Format entry field and enter the following formats so that your screen looks like Figure 1-20

TOTAL	ZZZ99.99
ACTUALPRICE	ZZZZ9.99
ITEMTOT	ZZZZ9.99

FIGURE 1-20
Altering Field Formats

Action Query Group Field Summary Text Report Parameter Help						
Field Settings						
Field Name	Data Type	Field Width	Display Format	Relative Position	Lines Before	Space Before
CUSTID	NUM	8				
ORDID	NUM	6				
SHIPDATE	DATE	9				
TOTAL	NUM	10	ZZZ99.99			
ORDID2	NUM	6				
ITEMDID	NUM	6				
PRODID	NUM	8				
ACTUALPRICE	NUM	10	ZZZZ9.99			
QTY	NUM	10				
ITEMTOT	NUM	10	ZZZZ9.99			

< | | | | | >

Enter the format mask (e.g. \$999.99)

Report Name: TwoQueryReport <Replace>

6. Press [Help] to learn about this field. Since the format you wish to learn about is a number format, press the [Scroll Down] cursor key until you reach the bolded "number format" in the text. Press [Select] to move to that topic and learn about number formats.
7. Press Return to quit the help system.
8. Go to Field Screen Three.

9. In the Align entry field of the CUSTID field, type Left and then press [Next Record]. Repeat this for all the remaining fields that are selected from the Ord table (ORDID, SHIPDATE, and TOTAL). Doing this causes those fields to print left-justified in the report.
10. Now move to the Skip entry field and enter an x beside the field that appears more than once ORDID2. You can identify multiple fields by looking at the field names. SQL*ReportWriter appends a number to a field name when more than one column with the same name is SELECTed in your report. In this example, ORDID is the first ordid column selected, ORDID2 is the second ordid column selected, etc.

FIGURE 1-21
Skipping Fields from
Report Output

Action Query Group Field Summary Text Report Parameter Help						
Field Settings						
Field Name	Align	Skip	Repeat	Function	Computed Value	Reset Group
CUSTID	Left					
ORDID	Left					
SHIPDATE	Left					
TOTAL	Left					
ORDID2		x				
ITEMID						
PRODID						
ACTUALPRICE						
QTY						
ITEMTOT						

< |

Do not print this field in the output.

Report Name: TwoQueryReport <Replace>

11. Exit this screen.

Editing Text

In this section you go to the Text Screen and edit the body text to change the layout of the report.

Select Text from the Main Menu.

Press [Next Record] several times until you see the screen with an Object of G_ORDTABLE and a Type of Body.

Move to the Text area by pressing [Next Field] several times.

Place the cursor on the first letter of the second field label (Shipdate) and press the Return key to place it on a separate line. Repeat this step for the TOTAL field.

Press [Insert/Replace] to put SQL*ReportWriter into insert mode.

Now line up the Total field reference (which is preceded by&) so your screen looks just like Figure 1-22.

FIGURE 1-22
Editing Text Format

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Text Settings								
Objectct: G_ORDTABLE			Type: Body			Status: Default		
Relative Position:		Repeat On Page Overflow: X		Justification: Left				
Lines Before:		Spaces Before:		Frequency:				
Width::								
Panel Number: 1				Text		Panels Defined: 1		
Order Id: &ORDID								
Shipdate: &SHIPDATE								
Total: &TOTAL								
IM								
Edit the text								
Report Name: TwoQueryReport <Normal> <Replace>								

- Execute your report. (See Figure 1-23.) It will display fields from the Ordtable query on the left side of your screen: their labels will be to the left of their values (from editing the text), and the values will be left-justified (from typing Left in the Align field). Fields from the Itemtable query will be displayed on the right side of your screen. Because you did not alter the default group positioning of the Itemtable query, you will notice that the format of those fields is the same as the first report you executed.

FIGURE 1-23
Finished Report

-ORDER SUMMARY-					
Customer Id: 100					
	<u>Itemid</u>	<u>Prodid</u>	<u>ActualPrice</u>	<u>Qty</u>	<u>Itemtot</u>
Order Id: 620	1	100860	35.00	10	350.00
Shipdate: 12-MAR-87	2	200376	2.40	1000	2400.00
Total: 4450.00	3	102130	3.43	500	1700.00
Order Id: 621		100861	45.00		450.00
Shipdate: 01-JAN-87	2	100870	2.00	100	280.00
Total: 730.00					
Sum			5180.00		
Customer Id: 101					
	Itemid	Prodid	Actualprice	Qty	Itemtot

Summary of Lesson III In Lesson III of this tutorial, you copied the report you built in Lesson II and learned how to:

- build a report with multiple sections, creating a parent/child relationship between the initial query, Ordtable, and the second query, Itemtable
- organize the two sets of data by specifying the layout of the two groups
- alter field default formats
- change the layout of one of the report sections by editing the body text

At this point, you can leave this tutorial with a completed report. The final lesson, however, will explain how to create two different types of fields wrap and computed. It also explains how to edit text by querying the text object you wish to edit. Finally, it will show you how to highlight fields and text. We suggest strongly that you complete Lesson IV of the tutorial.

Lesson IV: Creating a Wrap Field and a Computed Field

In this final lesson, you enhance the report you created in Lesson III by adding a wrap and a computed field. The following topics are covered in this section:

- modifying a query
- creating a wrap field
- changing the default report width
- creating a computed field
- editing text using the query mode
- using the text and field highlight feature.

The report you will produce is shown in Figure 1-24.

FIGURE 1-24
Break Report

-ORDER SUMMARY-					
Customer Id: 100 Name: JOCKSPORTS					
	Itemid	Prodid	Actualporice	Qty	Itemtot
Order Id: 620	1	100860	35.00	10	350.00
Shipdate: 12-MAR-87	2	200376	2.40	1000	2400.00
Total: 4450.00	3	102130	3.40	500	1700.00
Order Id: 621	1	100861	45.00	10	450.00
Shipdate: 01-JAN-87	2	100870	2.80	lee	280.00
Total: 730.00					
CUSTOMER TOTAL: 5180.00					
Customer Id: 101 Name: TKB SPORT					

Creating a Wrap Field

In this section you add a new field, the customer name, to the first query and then define it to be a wrap field. Note that you can make any field word-wrap; however, since the fields in this report are relatively small, you will add a longer field to the report so that the effect of word-wrapping will be more obvious.

Adding to the Query

1. Select Query from the Main Menu.
2. Move the cursor to the SELECT Statement area of the Ordtable query.
3. Add to the Query text so that your screen looks like Figure 1-25.

FIGURE 1-25
Adding a New Field to a Query

Query Name: ORDTABLE Query 1 of 2

SELECT Statement

```
SELECT NAME, ORCID, SHIPDATE, ORD.CUSTID, TOTAL
FROM ORD, CUSTOMER WHERE SHIPDATE > '31-DEC-86'
AND ORD.CUSTID = CUSTOMER.CUSTID
ORDER BY ORD.CUSTID
```

Parent-Child Relationships

Child Columns	Parent 1 Columns	Parent 2 Columns

Enter a SQL SELECT statement defining data for this report.

Report Name: TwoQueryReport <Replace>

Assigning a Group to the New Field

1. Select Field from the Main Menu.
2. To make the customer name appear with the customer ID, assign the Field Name, NAME, to the G_Customer group. Move the cursor to the Group entry field for NAME, and use the List of values to select G_Customer.
3. To make the NAME field label look like the CUSTID label, add a colon at the end of Name in the Label entry field.
4. The Field Width entry field is found on Field Screen Two. To see the field width, press [Scroll Right]. You can see that NAME has a width of 45 characters. Change the width to 10.
5. Press [Scroll Right] to move to Field Screen Three.
6. In the Align entry field for the NAME field, enter `Wrap`.

6. Press [scroll Right].
7. Enter a Display Format of ZZ9. 99% for the CUST_TOT field.
See Figure 1-27.

FIGURE 1-27
Entering a Display Format

Field Settings							
Field Name	Data Type	Field Width	Display Format	Relative Position	Lines Before	Spaces Before	
CUSTID	NUM	8					
NAME	CHAR	10					
ORDID	NUM	6					
SHIPDATE	DATE	9					
TOTAL	NUM	10	ZZZ99.99				
ORDID2	NUM	6					
ITEMID	NUM	6					
PRODID	NUM	6					
ACTUALPRICE	NUM	10	ZZZZ9.99				
QTY	NUM	10					
ITEMTOT	NUM	10	ZZZZ9.99				
CUST_TOT	NUM	10	ZZ9.99%				

< [] >

Enter the format (e.g. \$999.99).

Report Name: TwoQueryReport <List><Replace>

8. Press [Scroll Right] and then move to the Computed Value (Function) entry field.
9. Use the List of values to select %Total for the Computed Value (Function) for CUST_TOT. SQL*ReportWriter automatically determines a logical Reset Group. The Reset Group determines where the computed field value will be set to-. (For example, when computing total sales for individual departments, you would choose the group that owns the DEPTNO field, G_Ordtable, as the Reset Group. SQL*ReportWriter would then sum the first department's totals, reset to zero, sum the second department's totals, reset to zero, etc.) For this report, use the suggested Reset Group. See Figure 1-28.

FIGURE 1-28
Specifying a Reset Group

Action Query Group Field Summary Text Report Parameter Help						
Field Settings						
	Field Name	Align	Skip	Repeat	Computed Value	
					Function	Reset Group
^	CUSTID	Left				
	NAME	Wrap				
	ORDID	Left				
	SHIPDATE	Left				
	TOTAL	Left				
	ORDID2	Left	X			
	ITEMDID					
	PRODID					
	ACTUALPRICE					
	QTY					
v	ITEMTOT				% Total	G_ORDTABLE
	CUST_TOT					
<						
Enter the name of the group where computed field is reset to zero.						
Report Name: TwoQueryReport					<List><Replace>	

Editing Text

In this section you will go to the Text Screen and change the Summary field label and its default format. Because your report now consists of many text objects, you will query the text you want and then edit it. (Alternatively, you could press [Scroll Right] repeatedly, as you did earlier in this tutorial, until you see the Text Object you want to edit.)

1. Go to the Text Screen.
2. Press [Query].
3. Enter G_ORDTABLE in the Object entry field.

4. Press Next Field] and enter Footer in the Type entry field.
5. Press [Fetch].
6. Move to the Text area, delete the dashes, change the Sum label to CUSTOMER TOTAL: and then change its reference (&Customer_Total) position so that your screen looks like Figure 1-29.

FIGURE 1-29
Deleting and Pasting Text

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Text Settings								
Object: G_ORDTABLE			Type: Footer			Status: Default		
Relative Position:			Repeat On Page Overflow:					
Lines Before:			Justification: Left					
Spaces Before:			Frequency:					
Width:								
Panel Number: 1			Text			Panels Defined: 1		
^			CUSTOMER TOTAL: &Customer_Total					
v								
Report Name: TwoQueryReport				<Normal>		<Replace>		

Highlighting Text

In this-on you highlight a field, and its associated text.

1. Move to the Text field where the Object is G_CUSTOMER and the Type is Body, and then move your cursor to the "C" in CUSTOMER ID: in the Text field.
2. Press [Mark] and then move to the "N" in &NAME. The text that appears in reverse-video will be highlighted. Press [Highlight]. The Highlight Text List of values will appear as illustrated in Figure 1-30.

FIGURE 1-30
Highlighting Text with List of Values

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Text Settings								
Object: G_CUSTOMER			Type: Body			Highlight Text		
Relative Position: Lines Before: Spaces Before: Width:			Repeat On Page Overflow: Justification: Frequency:			Current Style Normal Underline Reverse Bold Underline/Reverse Underline/Bold Reverse/Bold Und/Reverse/Bold		
Panel Number: 1						Text		
^ Customer Id: &CUSTID Name: &NAME								
v								
Highlight the text with the last highlight style used.								
Report Name: TwoQueryReport						<Normal>		<Replace>

3. Select **Bold**. When you do so, you will see the text in boldface.
4. While you are on this screen, move your cursor to the N of the Name label and press Return to move it, and its reference, to the next line.
5. Exit this screen and execute your report. The report displays the wrapped, highlighted CUSTID and NAME fields in the control break group, and the CUST_TOT computed field with running percentages of the customer's purchases. See Figure 1-31. **(Note:** To see the CUST_TOT computed field, press [Window] once and then [Right] several times.)

**FIGURE 1-31
Finished Report**

-ORDER SUMMARY-

Customer Id: 100
Name: JOCKSPORTS

	Itemid	Prodid	Actualprice	Qty	Itemtot
Order Id: 620	1	100060	35.00	13	350.00
Shipdate: 12-MAR-87	2	200376	2.40	1000	2400.00
Total: 4450.00	3	102130	3.48	500	1700.00
Order Id: 621	1	100861	45.00	10	450.00
Shipdate: 01-JAN-87	2	100870	2.00	100	200.00
Total: 730.00					

CUSTOMER TOTAL: 5100.00

Customer Id: 101
Name: TKB SPORT

Summary of Lesson IV

In Lesson IV you enhanced the report you created in Lesson III and learned how to:

- modify a query by inserting text in the SELECT Statement entry field
- create a wrap field by setting the field alignment to Wrap
- create a computed field by creating afield, assigning a database field to it, and then assigning a function to it
- query and edit a text object
- highlight text using the Highlight List of values.

PART

II

**UNDERSTANDING
SQL*REPORTWRITER**

SQL*REPORTWRITER CONCEPTS

This chapter discusses the components of SQL*ReportWriter at a conceptual level. SQL*ReportWriter is a general purpose tool for developing and executing reports, specially designed for application developers who know the SQL language.

You can use SQL*ReportWriter to:

- combine multiple SQL statements in a single report to easily define complex relationships
- create ad hoc reports using a rich set of defaults
- perform complex calculations
- run reports interactively or in production environments with flexible runtime parameters
- fully customize all parts of the report definition.

SQL*ReportWriter uses Oracle's standard menu-driven, fill-in-the-form user interface to enhance your productivity and ease of learning.

Elements of a Report

A SQL*ReportWriter report is made up of the following objects

Queries define the data to be retrieved from the database.

Fields represent column expressions and report calculations from SELECT statements and describe how each is to be displayed.

Groups contain sets of fields. Groups are used to describe each section or subsection in the report and its relationships, as well as to dictate control breaks for subtotalling purposes.

Summaries are similar to fields, except they display subtotals and grandtotals rather than data direct from a query.

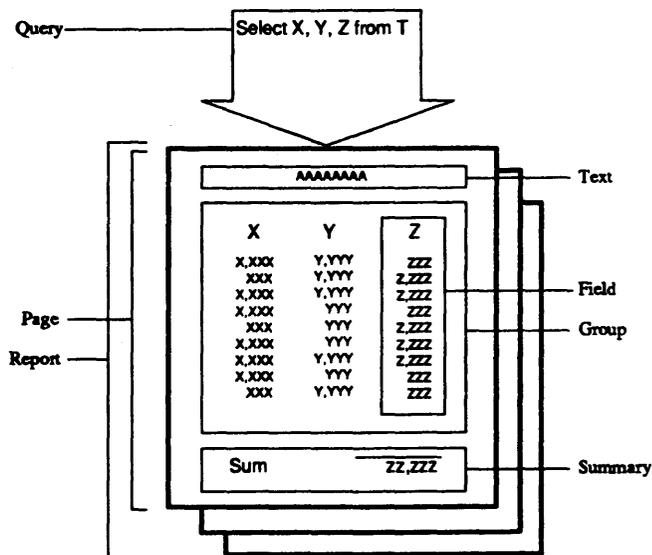
Texts contain fields, summaries, and parameter references, combined with literal strings, such as titles, and define the final report format.

The Report object defines the page size, margins, parameter form text comments, security, and history of the report.

Parameters contain literal values that you supply at runtime to control the behavior of the report. You can vary the data, routing of the output, and select national language settings.

These objects contain all the information needed to manipulate and produce reports. See Figure 2-1.

FIGURE 2-1
Report Objects



Every report contains at least one query, one group, and one field. Each object in a report has a set of attributes, or “settings”, which provides information about the object. For example, fields have settings for Width and Display Format, among others.

Most objects are initially created by default, and, in many cases, the default settings are sufficient. Simple forms are provided to edit the defaults and to create new objects, such as subtotals and grandtotals.

Report Building Process

There are a few simple steps involved in building a report:

- creating a blank report
- specifying the data and calculations to be used in the report
- specifying the format of the report.

SQL*ReportWriter separates the process of data access from report formatting, providing you with an additional degree of freedom.

Creating a Blank Report

In order to build reports successfully with SQL*ReportWriter, you must follow a few simple steps. First, select the Action choice from the Main Menu and then select New from the pull-down menu, and then enter a name for your report.

After you have created a blank report, it is helpful to enter a comment describing the purpose of the report, and at the same time define your page size and margins. Select the Report choice on the Main Menu to access these settings.

Specifying the Data

The next step is to define one or more queries. Queries enable you to specify the data you plan to use. You can access data from one or more tables residing in one or more databases. You can use multiple queries in a report, and you can create relationships between them.

Specifying the Report Format

Once queries are defined, you can use groups settings to specify where groups of data from your queries should be placed in your report; and/or specify control breaks (or master/detail relationships). You can think of groups as a tool to perform “coarse” or overall placement of data in your report. “Fine tuning” can be done using the field and text objects described below.

The next step in building reports is to modify the fields. For each column in the database that you selected, a field is created. By default, each field is made up of two parts: a label and the data of the field. You can change the default field label to any name, using any capitalization you wish; you can change the default data format to any format you desire.

The final step that is used to build a report is to modify the text. Using the text object you can change the positioning or appearance of the fields. You can move or highlight one, or any number of field labels. You can even highlight parts of field labels. You can also move or highlight one, or any number of fields. In the text object, you can add your own text to appear in a report. The text that you create is called "boilerplate text." You can reposition your boilerplate text, and any part(s) of it.

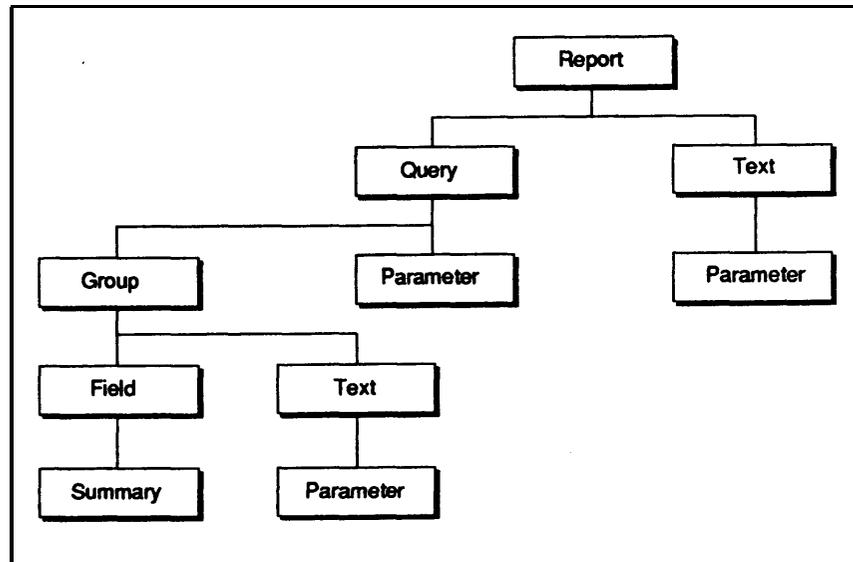
SQL*ReportWriter Objects

SQL*ReportWriter reports are built with the following object:

- Query
- Group
- Field
- Summary
- Report
- Text
- Parameter.

Every SQL*ReportWriter object (except Report) is owned by another object. For example, a group owns all of the fields within it. An object is not shared with another object; it has one, and only one, owner. Figure 2-2 shows the hierarchy of object ownership.

FIGURE 2-2
Object Hierarchy



The SQL*ReportWriter objects are described in the remainder of this chapter.

Query Objects

SQL*ReportWriter reports contain one or more query objects. Each query is made up of one SQL SELECT statement. SQL*ReportWriter allows you to enter each query directly. Each SELECT statement defines the rows and columns from specified tables or views which are to be used in the report, as well as defines calculations, sorting, or set operations, such as UNION.

Multiple Queries

Using SQL*ReportWriter, you can combine data from multiple SELECT statements in a single report. Because SQL*ReportWriter allows you to display unrelated data from two separate queries side by side, or establish relationships between them, you can create complex multi-part reports.

Multi-Part Unrelated Queries

Sometimes reports contain parts of data that are completely unrelated. To build these reports, specify the queries without specifying any relationship. In this case, SQL*ReportWriter fetches the data for each query in turn, without matching data across queries. You might use this approach for a report containing a list of products and a separate list of customers, as in Figure 2-3.

FIGURE 2-3
Multi-Part Unrelated Queries

① Products

② customers

ProdId	Name	CustId	Name
100060	ACE TENNIS RACKET I	101	TKS SPORT SHOP
100061	ACE TENNIS RACKET II	102	UOLLYRITE
100070	ACE TENNIS BALLS-3 PACK	103	JUST TENNIS
100071	ACE TENNIS BALLS-6 PACK	104	EVERY MOUNTAIN
100090	ACE TENNIS NET	105	K + I SPORTS
101060	SP TENNIS RACKET	106	SHAPE UP
101063	SP JUNIOR RACKET	107	WOMENS SPORTS
200230	RH: "GUIDE TO TENNIS"		
200376	SB ENERGY BAR-6 PACK		
200380	SB UITA SNACK-6 PACK		

Related Queries

In some reports, the data in one part is related to the data in another part. For example, in Figure 2-4 each Order Id and Customer have related Items, Products, and Amounts. This relationship is termed a "parent/child," or "master/detail," relationship. For each query in a report, you can designate another query as its parent, and indicate the columns in the parent which match corresponding columns in the child. Then, when the report is run, the child query will be re-executed for each row of the parent, retrieving only matching rows.

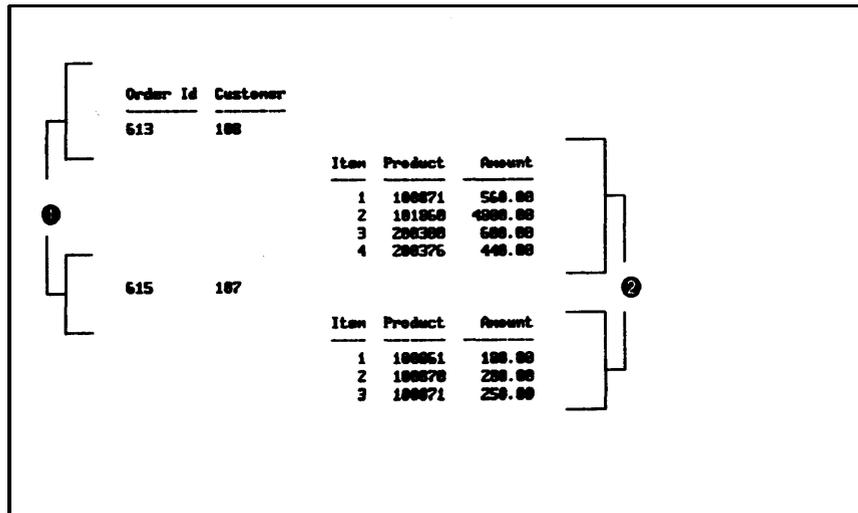
Parent/child relationships are used to create control breaks in a report, displaying one or more rows from the child for each row of the parent (-Figure 2-4). You can establish these relationships in two different ways using SQL*ReportWriter: 1) by using two queries; or 2) by using one query and an additional, manually created Group.

- To create a relationship between two queries, specify one query as the parent of the other.
- To create a relationship between fields of the same query, insert a Group and assign an appropriate field to it. See "Inserting a Group" in Chapter 5 of the SQL*ReportWriterReference Manual for details.

A simple example of relating two queries can be found in the tutorial in Chapter 1, in which a detail query is nested within a master query. Separate queries retrieve the data for each section of the report. Another example is shown below in Figure 2-4, where a query against the Item table is nested within a query against the ORD table.

FIGURE 2-4
Related Queries

- ① ORD(parent)
- ② ITEM (child)

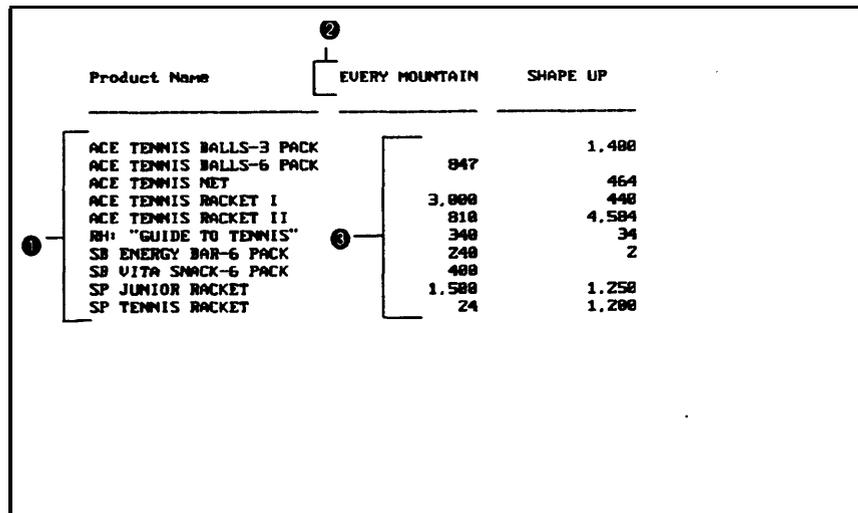


Matrix Reports

You can define a second parent for a query to define a matrix report. Each query, like all other objects in a report, is named. (In the rest of this book, queries will be referred to by name, rather than by the data they retrieve.) In the matrix report in Figure 2-5, Q_Prod and Q_Cust are defined as the two parents of Q_Ord. Q_Prod provides the rows of the matrix, Q_Cust the columns, and Q_Ord the cells. The values of Q_Ord must match the values of both Q_Prod and Q_Cust in order to be displayed. See the section on “Group Objects” later in this chapter for more details.

FIGURE 2-5
Matrix Report with Three Queries

- ① Q_Prod
- ② Q_Cust
- ③ Q_Ord



Parameters

SQL*ReportWriter enhances production reporting by allowing you to parametrize queries and text objects, and also supply values for the parameters when you run the report. Parameters can be used to replace expressions in WHERE, GROUP BY, ORDER BY, HAVING, CONNECT BY, and START WITH clauses of queries. There are two types of parameters in SQL*ReportWriter: bind and lexical.

Bind parameters and lexical parameters are handled differently by SQL*ReportWriter. With bind parameters, one value, or word, is substituted into the parameter reference; with lexical parameters several values, or words, may be substituted into the parameter reference.

Bind parameters cause

- the query to be parsed (checked for errors) as soon as you enter it in SQLREP
- the values to be substituted for the bind parameters each time a new record is fetched from the database (while executing the report using either SQLREP or RUNREP).

Thus, bind parameter values are changed throughout the execution of a report. Note that queries are not parsed again during run-time for bind parameters.

Lexical parameters cause

- the values to be substituted for the lexical parameters only once, before the query is parsed
- the query to be parsed just before the report is executed.

Thus, lexical parameter values are not changed throughout the execution of a report while using either SQLREP or RUNREP. Note that queries are parsed during run-time for lexical parameters.

See Chapter 10 in the SQL*ReportWriter Reference Manual for more details.

To reference a bind parameter in a query, you simply add it to your SELECT statement. You must place a colon(:) before the bind parameter name. For example

```
SELECT * FROM ORD
WHERE ORDNO = : ORDPARAM
```

When executing the report containing the above query, you can supply the bind parameter value for ORDPARAM in one of three ways, depending on how you run the report using RUNREP. You can

- Supply a default value in the report definition which is used if you do not provide a value when the report is run.
- Enter the parameter value on a form which appears when you run the report interactively.
- Enter the parameter value on the command line when you run the report.

You can reference lexical parameters as elements of expressions in the following clauses of your SELECT statement: WHERE, GROUP BY, HAVING, ORDER BY, CONNECT BY, and START WITH.

To reference a lexical parameter in a query, you must first create the parameter and assign default values to it in the Parameter Screen. You must place an ampersand (&) before the lexical parameter name. For example

```
SELECT * FROM ORD
WHERE & ITEMS
```

When executing the report containing the above query, you can supply the lexical parameter value for ITEMS in one of three ways, depending on how you run the report using RUNREP. You can

- Supply a default value in the report definition which is used if you do not provide a value when the report is run.
- Enter the parameter value on a form which appears when you run the report interactively.
- Enter the parameter value on the command line when you run the report.”



Field Objects

Fields may be thought of as containers. These containers display values derived either from your queries, SQL*ReportWriter functions, or user exits. When you enter a SELECT statement, SQL*ReportWriter generates a field for each of the items in the SELECT list. The default definitions of these fields, such as DataType and Width, come from the data dictionary of the DBMS. (DataType and Width are field settings; see Chapter 6 in the SQL*ReportWriter Reference Manual for more details.)

Field Settings

Field settings give you explicit control over the structure, position, and format of each field. Field settings are located on the Field Screens, and fall into three categories structural settings, positioning settings, and formatting settings.

Structural Settings

Structural settings define the source of the values of a field or how the field is calculated. For example

- the Source setting shows the name of the column in the SELECT statement that defines the field. This setting also provides the field on which to base a computed field (see “Computed Fields” later in this chapter).
- the Function setting specifies the type of calculation used to calculate the value of a computed field. Fourteen standard functions are available and discussed in the “Computed Value (Function)” section in Chapter 6. Should you wish to use a non-standard function, you can write a third generation language-based program (e.g., C, COBOL, FORTRAN) and then access it with a user exit. (See Appendix E in the SQL*ReportWriter Reference Manual for details.)

Positioning Settings

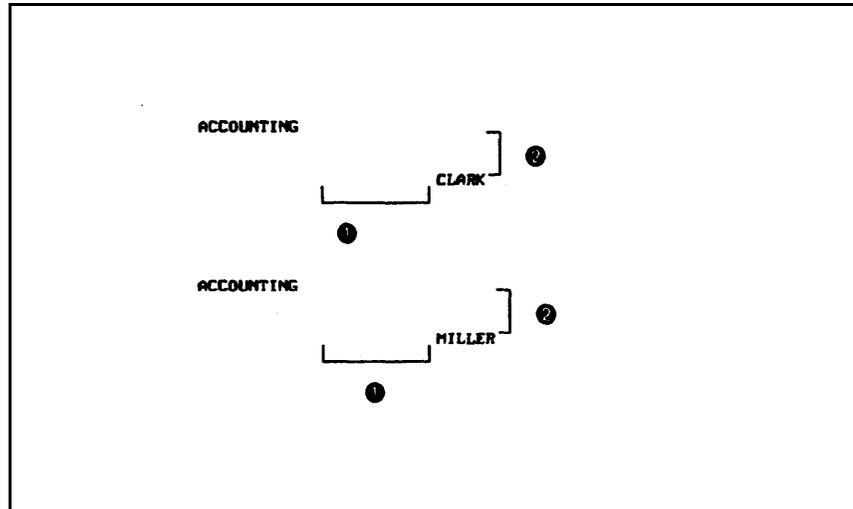
Positioning settings define the location or position of each field in relation to the prior field. For example:

- the Relative Position setting tells SQL*ReportWriter to place the field either to the Right of or Below the previous field.
- the Lines Before and Spaces Before settings tell SQL*ReportWriter to insert the specified number of lines and spaces, respectively, before outputting the field. See Figure 2-6.

FIGURE 2-6
Field Positioning

① Spaces Before = 10

② Lines Before = 3



Formatting Settings

Formatting settings determine the appearance of a value in the field. For example

- the Display Format setting is a string that identifies how to format date and numeric values
- the Align setting allows you to justify the values of a field to the left, right, or center of the field's boundary. You can also have wrap- and variable-width fields. Wrap causes text to word-wrap within a field's boundary. *Variable* causes a field's boundary to be the length of the field's value; that is, trailing spaces in the field's value are truncated.

Computed Fields

Computed fields are manually created fields that show the results of calculations performed on values of their fields after the values have been retrieved from the database. They provide significant functionality that is not available by way of SELECT statements alone, and also can reduce the complexity or number of queries in your reports. For example, you can use computed fields to compute each employee's salary as a percent of the department total, or to compute the running total of salaries.

To create a computed field, enter a new Field Name along with the other attributes that are necessary to define a field. Default definitions of computed fields, such as Data Type and Width, are inherited from the field's Source setting. For Source settings, you can specify a SELECT column name, a system variable (&DATE, &PAGE, &NUM_PAGES for calculating and displaying dates and page numbers), a user exit (to invoke any procedural language program), or &SQL (for any SQL statement). Restrictions may apply in some of these cases. Refer to Chapter 6 of the SQL*ReportWriter Reference Manual for details.

The last step needed to create a computed field is to specify which default Function performs the calculation, and where the calculation is reset to zero. See Figure 2-7.

**FIGURE 2-7
Computed Field Report**

① Computed Field

Order Id	Customer	Item	Product	Amount	% Total
613	100	1	100071	560.00	8.75
		2	101060	4000.00	75.00
		3	200300	600.00	9.30
		4	200376	440.00	6.00
615	107	1	100061	100.00	25.35
		2	100070	200.00	35.44
		3	100071	250.00	35.21



Highlighting Fields There are four ways in which to highlight fields

- highlighting all fields of a group (on Group Screen Three)
- highlighting fields, text, or parts of text using the Highlight List of Values (on the Text Screen)
- highlighting fields and text by embedding printer control codes (on the Text Screen)
- conditionally highlighting fields using user exits (on Field Screen One).

For details about highlighting fields of a group, see “Highlight” in Chapter 5 of the SQL*ReportWriter Reference Manual.

For highlighting fields, text, or parts of text using the Highlight List of Values or using printer codes, see “Highlighting Text” in Chapter 8 of the SQL*ReportWriter Reference Manual, or see “Highlighting” in the “Text Objects” section later in this chapter.

On the Field Screen, user exits enable you to conditionally highlight fields. For example, if you want to bold all negative dollar amounts, you can create a user exit and pass the appropriate printer code back to the SQL*ReportWriter field. For more information, see “Source” in Chapter 6 and “Conditionally Highlighting Fields” in Appendix E of the SQL*ReportWriter Reference Manual.

Group Objects

A group is a set of fields either derived automatically from a query or created manually by assigning one or more fields to it. A group has two primary functions in a report. The first is as a visual section or subsection; the second is as a logical control break which occurs each time a new set of data for the group is found. Groups let you:

- position one group in relation to other groups, providing “section-at-a-time” layout
- use the control break for subtotalling and pagination purposes
- create specialized layouts such as matrix reports, form letters, and multi-column mailing labels.

Managing Groups

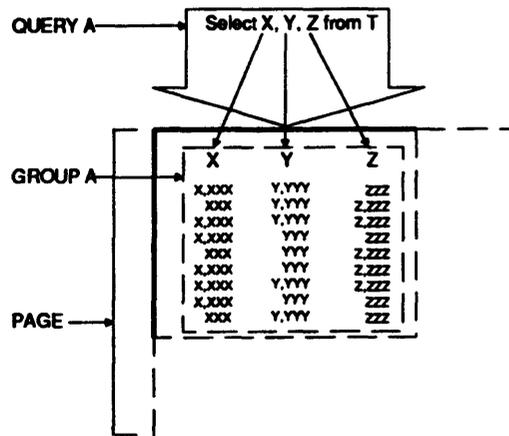
SQL*ReportWriter creates a default group for each query you enter. This default group contains all the fields from the query. When you create additional groups for a query, you must order them according to the nesting you want and assign one or more fields to each group.

Groups Represent Queries

Each group is associated with one query and contains one or more fields. Groups are nested according to the hierarchical relationship between them. Default groups, which are created automatically from queries, are nested according to the parent/child relationships of their queries.

Thus, if a query named Q_Ord is the parent of a query named Q_Item, the G_Item group will be nested within the G_Ord group. You will be able to subtotal the records from the G_Item group each time a new record of the G_Ord group is found. Figure 2-8 shows the default relationship of queries and groups.

FIGURE 2-8
Relationship of
Groups and Queries



Multiple Groups per Query

You can separate the fields from a single query into multiple groups to create additional nested sections. This allows you to create multi-level reports with only one query. To separate the fields, you create a new group and then assign one or more fields to the new group. The hierarchy is created by the order in which the groups are entered on the Group Screen.

Records

In the same way that groups represent queries, records represent rows of data fetched from the database. Each time you create a new group, an implicit record definition is created consisting of the fields in the group.

Group Settings

Various settings control the behavior and appearance of groups. The settings appear on the three Group Screens.

Positioning and Spacing

Each group has a setting called Relative Position which indicates where the group is to appear relative to the prior group, either a parent or a sibling. If you set the Relative Position of G_Item to *Right*, the fields of G_Item appear to the right of those in G_Ord. If you change the setting to *Below*, the entire group is repositioned to the area immediately below G_Ord.

The Lines Before and Spaces Before settings allow you to insert blank lines and spaces between groups. The Record Spacing setting allows you to control the number of blank lines between records of the group. Other settings provide default positioning and spacing for the fields in the group. In Figure 2-9 the Relative Position of G_Item is *Right*, and the Lines Before and Spaces Before settings are 3 and 10, consecutively.

FIGURE 2-9
Positioning Groups

- ① Lines Before=3
- ② Spaces Before=10

OrdId	Ship Date	Product	Quantity
604	30-JUN-05		
		ACE TENNIS RACKET	174
		ACE TENNIS RACKET II	84
		ACE TENNIS RACKET I	408

Print Direction

This setting lets you control where the second and subsequent records of each group appear in relation to the first. The default value, Down, places each record below the previous record. The other settings place each record to the right of the previous record (Across), form the cells of a matrix (Crosstab), or squeeze more rows onto a page by wrapping back to the top or left margin instead of starting a new page (Down/Across and, conversely, Across/Down). See Figure 2-10.

FIGURE 2-10
Print Direction

- ① Down
- ② Down/Across
- ③ Across
- ④ Across/Down

①	Smith Jones Rose James Bond	②	Smith James Jones Rend Rose
③	Smith Jones Rose James Bond		
④	Smith Jones Rose James Bond		

Pagination

The Page Break setting controls when page breaks are generated. The default setting (blank) starts anew page only when there is no more room on the page for more records of the group. A value of Always creates a new page for each record of the group. A value of Condition creates anew page whenever there is insufficient room on the page for both the current record of the group and for all the records belonging to its descendent groups.

Another setting that impacts pagination is Multi-Panel, which controls what happens when there is not enough room on the current page to format one of the fields in the group. If Multi-Panel is selected, the field, and those after it, appear on a new panel. If the setting is left blank, SQL*ReportWriter attempts to wrap around and place the field on the same panel.

In this sense, SQL*ReportWriter supports “wide” reports as well as “long” reports. A wide report is a report in which not all of the fields appear on the same page. In this case, the extra pages needed to hold all the fields are called “panels,” to distinguish them from the extra pages needed to hold additional records. See Figure 2-11.

FIGURE 2-11
Wide Report

<u>Ord</u>	<u>ItemID</u>	<u>ProdID</u>	
610	1	100860	
614	2	100861	<u>ItemTot</u>
611	3	100671	99
623	4	100878	16
			83
			25

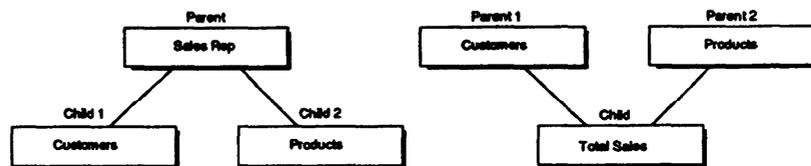
Page 1

Page 2

Paths/Hierarchy

It is useful to think of the nesting of queries and groups as moving along paths from child to parent to grandparent. As in a family tree, a parent may have many children. This sort of relationship is used to create master/detail/detail reports. Since there are no restrictions on how many children a parent can have, or how “tall” a “family tree” is, you can build nested detail sections to an unlimited level in a report. Conversely, assigning two parents to a child enables you to build matrix, or crosstab, reports. See Figure 2-12.

FIGURE 242
Data Paths/Hierarchy



Matrix Reports

The layout for matrix, or crosstab, reports can be created easily using the Print Direction and Matrix settings. Matrix reports require three queries, structured as described earlier (two parents, one child). The first query represents the rows, the second the columns, and the third the cells of the matrix. To define the layout, set the Print Direction for the groups for the column, row, and cell queries to Across, Down, and Crosstab, respectively. Also, enable the Matrix setting for each group. See Figure 2-13.

FIGURE 2-13
Matrix Report

- ❶ Print Direction =Down
- ❷ Print Direction=Acroas
- ❸ Print Direction= Crosstab

Product Name	EVERY MOUNTAIN	SHAPE UP
ACE TENNIS BALLS-3 PACK		1,400
ACE TENNIS BALLS-6 PACK	847	
ACE TENNIS NET		464
ACE TENNIS RACKET I	3,000	440
ACE TENNIS RACKET II	810	4,584
RH: "GUIDE TO TENNIS"	340	34
SB ENERGY BAR-6 PACK	240	2
SB VITA SNACK-6 PACK	400	
SP JUNIOR RACKET	1,500	1,250
SP TENNIS RACKET	24	1,200

Summary Objects

A summary is a special kind of field object. Summaries are used to calculate subtotals, running totals, and grand totals. The value of a subtotal is displayed and reset to zero periodically throughout a report. A running total is different because the intermediate totals between reset points are displayed; a grand total's value is never reset. See Figure 2-14.

FIGURE 2-14
Summary Report

- ❶ Summary

Ordid	Predid	Total
613	100071	500
	200300	600
	101050	4000
	200376	400
615	100051	100
	100071	250
	100070	200
Grand Total		7110

You can summarize any field in a report, including computed fields. Summaries can be calculated and displayed at any level of your report.

Print Group

The Print Group defines where the summary will be placed in the report output. You can place a summary in a group Header, Footer, or Subfooter. By default, the summary will be placed in the Footer of the group containing the field being summarized. A group footer is a text object that appears after each set of records in a group (see "Text Objects" later in this chapter for more information). In matrix reports, the summary is placed in the Subfooter. See Figure 2-15 for an illustration of print groups.

FIGURE 2-15
Print Group

- ① Print Group= Report
- ② Print Group= G_ORD

Ordid	Prodid	Total	Ordid	Prodid	Total
613	100071	500	613	100071	500
	200300	600		200300	600
	101060	4000		101060	4000
	200376	400		200376	400
			Grand Total	7110	②
615	100061	100	615	100061	100
	100071	250		100071	250
	100070	200		100070	200
			Grand Total	7110	②
		Grand Total	7110		①

Reset Group

A Reset Group is defined for each summary and controls how often the value is setback to zero. The value resets back to zero for each record in the reset group. By default, each summary is reset at the group directly above the group that contains the field being summarized. Thus, in Figure 2-15, the default Reset Group for the summary based on ITEMTOT is G_Ord. You may also specify REPORT or PAGE as a reset group to compute a grand total or a page total.

FIGURE 2-16
Reset Group

- ① Reset Group = G_ORD
- ② Reset Group = Report

Ordid	Prodid	Total	
613	100071	500	
	200300	600	
	101060	4000	
	200376	400	
	Subtotal	6400	①
	Grand Total	7110	②
615	100061	100	
	100071	250	
	100070	200	
	Subtotal	710	①
	Grand Total	7110	②

Summary Functions

The Function for a summary determines the calculation to use on a field. There are two types of functions: standard (those provided by SQL*ReportWriter), and non-standard (those you create yourself). Standard summary functions are:

- *Sum*
- *Min*
- *Max*
- *Count*
- *Avg*
- *%Total*
- *First*
- *Last*.

Each function, except First and Last, has a variation for running summaries, referenced by prefixing the function name with R_, as in R_Sum (see "Running Functions" later in this section). The non-running versions of the functions (e.g., Sum, Min) are known as periodic functions. Periodic functions accumulate a value until a breakpoint is reached, print the result, and begin again.

Non-standard functions are programs that you write yourself in a programming language. From SQL*ReportWriter, you may access your programs with a user exit. For more information, see Appendix E in the SQL*ReportWriter Reference Manual.

Running Functions

In order to compute running totals, select a running function instead of a periodic function. Running functions display intermediate values and are useful for running totals and ranking (running count). The accumulating value will be displayed for each record of the Print Group, and reset to zero for each record of the Reset Group.

Forward Referencing

The automatic forward referencing capability in SQL*ReportWriter lets you display summary data prior to the detail data which it summarizes. One example of forward referencing is placing a summary in the Header of a group rather than the Footer. Another example is shown earlier in Figure 2-15, where the grand total of sales is displayed next to each order on every page. This is achieved by setting the grand total's Reset Group to Report and the Print Group to G_Ord.

Row, Column, and Comer Summaries

For matrix reports, you can create row, column, and comer summaries of values in a matrix. For row summaries, select the group with a Print Direction of Down as the Print Group. The Print Group for column summaries is the group with a Print Direction of Across. Lastly, the Print group for corner summaries is Report. Note that the comer summary sums the column summary. The Reset Group for these summaries is the same as the Print Group. This becomes clear if you note that anew column total should appear each time anew record of the Across group is found, and a new row total should appear for each record of the Down group. See Figure 2-17.

FIGURE 2-17
Row, Column, and Comer Summaries

- ① Column Summary
Print Group=Down (sub-footer)
- ② Corner Summary
Print Group=Report (cell's footer)
- ③ Row Summary
Print Group=Across (sub-footer)

Product Name	EVERY MOUNTAIN	SHAPE UP	Sun
ACE TENNIS BALLS-3 PACK		1,400	1,400
ACE TENNIS BALLS-6 PACK	847		847
ACE TENNIS NET		464	464
ACE TENNIS RACKET I	3,800	448	3,448
ACE TENNIS RACKET II	810	4,584	5,394
RH: "GUIDE TO TENNIS"	340	34	374
SB ENERGY BAR-6 PACK	240	2	242
SB UITA SNACK-6 PACK	400		400
SP JUNIOR RACKET	1,500	1,250	2,750
SP TENNIS RACKET	24	1,200	1,224
	7,161	9,374	16,535

Formatting Summaries

By default, a summary inherits its `DataType` and `Width` from the field being summarized. The Summary Screen provides settings for overriding these defaults, so a summary can appear with attributes different from the field it summarizes.

Text Objects

Most reports cannot be considered complete without page headers and other textual information, including page numbers and often references to report data. Text objects are physical areas in your report filled with boilerplate text and references to fields, summaries, system variables, printer codes, and parameters. Default versions of some text objects are generated by SQL*ReportWriter as you create queries and groups; others must be created manually.

Editing text objects gives you the ultimate control needed for highly formatted reports. By editing these objects, you can override the positioning controls on the Field Setting Screens. You can also add additional text, and customize the column headings to create reports that exactly meet your needs.

Report and Page Text Objects

Report and page objects are associated with the report and page as a whole, are not dependent on the groups and fields in the report, and are not created by default. There are six types of Report and Page text objects. See Figure 2-18.

Report Title Page — Appears once at the beginning of the report on a set of separate pages.

Report Header — Appears once at the beginning of the report on the same page as the first data in the report.

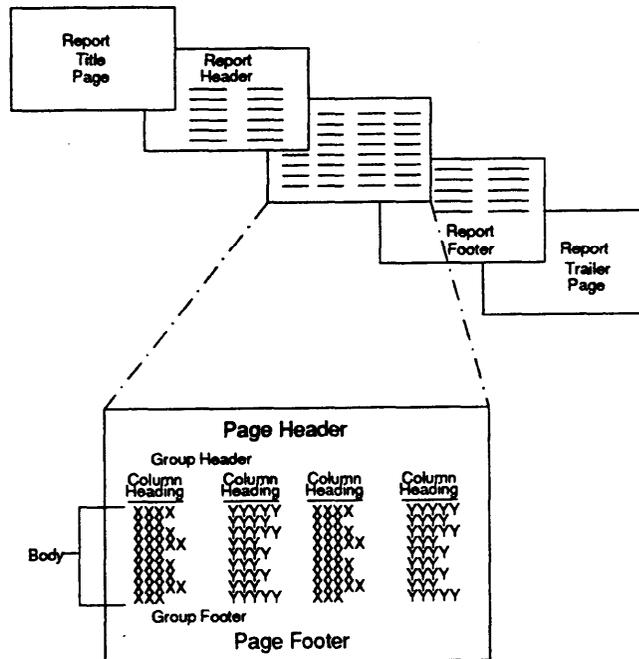
Page Header — Appears at the top of each page, except the Title and Trailer pages.

Page Footer — Appears at the bottom of every Page, except the Title and Trailer pages.

Report Footer — Appears once at the end of the report on the same page as the last data in the report.

Report Trailer Page — Appears once at the end of the report on a set of separate pages.

FIGURE 2-18
Text Objects



Group Text Objects

Group text objects are associated with each group in the report. These objects contain the fields, summaries, and column headings for each group. SQL*ReportWriter has five types of group text objects.

The rules for groups are somewhat more complicated than those for report and page objects. The descriptions that follow apply to the first group in a report. The variations used to format other groups are described more fully in Chapter 5 of the SQL*ReportWriter Reference Manual.

- Group Header—Appears before each set of records in the group.
Group headers are not created by default and are used to add text that describes the set of records as a whole.
- Group Footer—Appears after each set of records in the group.
SQL*ReportWriter normally places summaries in the footer of the group containing the field to be summarized. This is done so that summaries can be aligned easily with the field being summarized.
- Column Heading—Normally contains the Field Labels for each field. By default, SQL*ReportWriter aligns each label with the corresponding field in the Body.

- Body-Usually appears after the Column Heading and repeats once for each of the records in the group. The Body normally contains the fields that belong to the group. If the Label Position for the group is set to Left, the Body contains the field labels placed to the left of each field as well. The fields are positioned within the Body according to the settings on the Group and Field screens.
- Sub-footer-Exists only in matrix reports. A Sub-footer appears after the values of the child group that makeup the cells of the matrix. The Sub-footer of the Down group contains row summaries The Sub-footer of the Across group contains column summaries.

See Figures 2-19 and 2-20 for an illustration of Group text object.

FIGURE 2-19
Group Text Objects

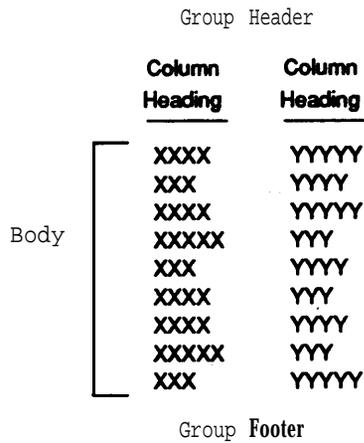
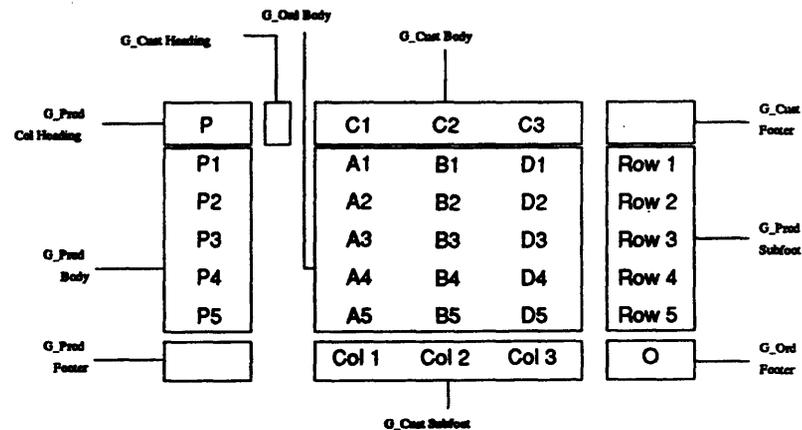


FIGURE 2-20
Matrix Groups



Editing Text Objects

As you add groups, fields, and summaries to your report SQL*ReportWriter creates default versions of the text objects as needed. For example, when you add a group, default Body and Column Heading objects are created; when you add summaries, default Group Footers appear as well. Once default versions of these objects exist, you can edit them manually to fine-tune the appearance of your report.

Even after you have edited these objects, SQL*ReportWriter allows you to continue making positioning changes to individual Fields, Groups, and Summaries on their respective screens. These changes, however, are not reflected in the text objects or report output. To do so would cause your carefully made edits to be discarded. To reflect your changes in the text, delete the edited text object. SQL*ReportWriter will automatically create anew default text object reflecting your changes.

The Status setting on the Text screen shows "Default" when a particular object and its type is unmodified, and shows "Edited" when you have manually modified a particular object and type. For example, if you modify the G_CUST body, its Status will be "Edited." All remaining G_CUST objects, for example the G_CUST header, will still have the Status of "Default." It is usually a good idea to edit group text objects only after you have created all the groups, fields, and summaries needed for the report, and have approximated the final layout with the positioning controls on the Field Screens.

You can easily restore the default versions of the text objects by simply deleting the edited panels of the object; SQL*ReportWriter will regenerate the default versions the next time you select the Generate or Execute actions, or when you re-enter the Text Screen.

Referencing Fields and other objects

Text objects can reference fields and summaries, as well as system variables, printer codes, and user-defined parameters. These Objects are referenced by putting an ampersand (&) in front of the name of the object. In Figure 2-21, the SHIPDATE field is referenced as &SHIPDATE.

FIGURE 2-21
Referencing Fields

- ❶ Field Labels
- ❷ Field References

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Text Settings								
Object: G_ORDTABLE		Type: Body			Status: Default			
Relative Position:		Repeat On Page Overflow: X						
Lines Before:		Justification: Left						
Spaces Before:		Frequency:						
Width:								
Panel Number: 1					Panels Defines: 1			
^ Order Id: &ORDID Shipdate: &SHIPDATE Total: &TOTAL ❶ ❷ v								
Edit the text.								
Report Name: Two Query Report			<Normal>			<Replace>		

Similarly, you can reference the system variables for current page number, number of pages in the report, and current date by &PAGE, &NUM_PAGES, and &DATE respectively. The values of system variables are maintained automatically by SQL*ReportWriter.

You can also directly reference a printer code by putting an ampersand before the printer code number. For example, to reference printer code #1 in your printer description file, you would types &1 in the text object.

Some restrictions limit which fields and summaries can be referenced in the various kinds of text and query objects. These limits occur when you try to reference a field in a text object, but a unique value for the field cannot be determined. There are no restrictions on referencing system variables and user-defined parameters.

Panels

SQL*ReportWriter can accommodate reports that are wider than a single page. This is useful when, based on the positioning logic you have specified, there are more fields in a single record of a group than can be placed on a single sheet of paper.

When this occurs, SQL*ReportWriter divides the associated text objects into multiple chunks, called panels. Default panels are generated automatically. You can explicitly move fields onto new panels by specifying a Relative Position of Panel for them on Field Screen Two.

You can create additional panels for Report and Page objects by inserting panels for them on the Text Screen. After inserting the panel, give it an appropriate number and add the necessary text. You can create additional panels for Group text objects in the same way.

Pages

SQL*ReportWriter distinguishes between the need to place all fields in a group on one page and the need to place all the records in a group on the same page. The first condition generates additional panels as described above. The second condition generates additional pages, depending on how many rows exist for each of the groups in the report.

New panels are not created based on the number of records selected. Thus, you can always tell how many panels there are in a report (that is, how wide it is) without running it. In order to know how many pages there are, the data must actually be retrieved from the database.

Positioning

Text objects are positioned using the same mechanisms used for fields and groups; that is, Relative Position, and Lines and Spaces Before. The Repeat on Page Overflow setting allows you to repeat “break text” and /or column headings on overflow pages.

Size and Justification of Text Objects

Each text object occupies some number of lines and spaces in the report output. The rules for determining the default dimensions vary by text object and depend on both the actual contents of the object, and whether the object has been edited.

By default, report and page text objects areas wide as the number of spaces between the right and left margin, and as long as the area between the top and bottom margins. You can change the default margin settings on the Report Screen. (See Chapter 9 in the SQL*ReportWriter Reference Manual).

For group objects, the default Print Direction, Relative Position, and the sizes of the individual objects (fields, text, etc.) in the group determine the dimensions. You can change the Print Direction, Relative Positions, and spacing defaults on the Group Screens. (See Chapter 5 in the SQL*ReportWriter Reference Manual for details.)

The default dimensions for all text objects of a group are equal to the dimensions of the largest object of the group, based on the text and fields of each text object. You can change the default settings in two ways on the Text Screen 1) by changing the default Lines Before, Spaces Before, and Width; and 2) by adding and deleting extra spaces as needed to attain the desired layout.

Each line of text in an object obeys the Justification setting individually. Lines are left- or right-justified, or centered within the width of the text object. Note that justification of Center or Right is disabled for Matrix Groups and groups with a Print Direction of Across. To align the across fields, use the Align setting on Field Screen Three.

Highlighting

SQL*ReportWriter supports two ways to highlight text in your reports that are displayed on the screen or on hardcopy:

- using text attributes that are already created for you
- using text attributes that your SQL*ReportWriter administrator can create.

The highlighting attributes that are already created for you are Bold, Reverse Video, Underline, and any combination of these. To highlight text, you simply mark the text you want to highlight, and then select the appropriate style from a List of values. Your SQL*ReportWriter administrator can create any number of highlighting attributes that are supported by your printer. To highlight text using these customized highlighting attributes, simply place the cursor before the text or database field you wish to highlight and enter a printer control code (such as, &2).

Note: If you use customized highlighting attributes, you will only be able to see the highlight when it is printed. Customized highlighting will not be displayed on your screen.

The Report Object

The Report object has a number of settings used to control overall page layout, parameter form defaults, and security. You can modify those settings, enter comments about your report, and view report history.

Page Layout

The report page size is identified by the Page Length and Page Width, and the Margins identify the number of lines and spaces to leave blank for the page borders. Spacing controls on other objects operate within the confines of the page size and margins.

Parameter Form Defaults

You can change the appearance of the Run-time Parameter Form, modifying the Title, Hint, and Status default lines by editing the Report object. You can also add comments to your report, keying in a description of your report's purpose to assist other users, by editing the Report object.

Security

SQL*ReportWriter provides report-level security. By adding names to, and deleting names from the Access List, you are able to grant and revoke privileges of other users for your report.

History

Each report has a report history displayed on the Report Screen. The History specifies the username, date, and version number for the time at which the report was created and last modified.

Parameter Objects

As discussed earlier in this chapter, bind and lexical parameters allow you to supply values that modify the query at the time you run the report. However, parameters also serve other roles. You can directly reference parameters in text objects, as though they were fields, and you can use system parameters for several purposes. There are seven system parameters. DESTYPE, DESNAME, DESFORMAT, and COPIES enable you to set report destination information. CURRENCY, THOUSANDS, and DECIMAL enable you to specify which symbols should be used for dollar amounts in your report.

DESTYPE allows you to specify Screen, Printer, File, Sysout, or Mail as the type of output. DESNAME allows you to specify the name of a file or printer to which the output should be written. DESFORMAT indicates the printer description file; that is, the file that identifies special characteristics about the printer receiving the report output. COPIES indicates the number of copies to be printed when the DESTYPE is Printer. For more information about defining your own printer drivers for use by SQL*ReportWriter, see Appendix H in the SQL*ReportWriter Reference Manual. For more information about system parameters, see Chapters 1 and 2 in the SQL*ReportWriter Reference Manual.

Parameters referenced in queries receive default definitions which you can modify on the Parameter Screen. You can define or modify the Data Type, Width, Default Value, prompt (Label), Input Format, and Output Format for each parameter.

You can also enter parameters manually for subsequent references in queries and text objects. This is convenient if, for example, you wish to display the name of the person running the report on the Title page.

Run-time Parameter Form

The Run-time Parameter Form contains a list of parameters that you wish to be displayed during run-time. You can request that the Form appear at run-time, giving you the opportunity to enter a value for each parameter. The name you specify on the Parameter Screen is used to prompt for values on the Run-time Parameter Form. When the report is run, the value entered on the Form must match the Data Type and Width that is specified on Parameter Screen One. You can choose which parameters will not appear on the Run-time Parameter Form by using the Skip option on Parameter Screen Two. (See Chapter 10 in the SQL*ReportWriter Reference Manual for details.)

Elements of SQL*ReportWriter

SQL*ReportWriter consists of several programs which are described briefly below. For more details, see the SQL*ReportWriter Reference Manual.

SQLREP

The SQLREP program allows you to define and run reports using menus, fill-in-the-form screens, and a comprehensive on-line help system. SQLREP stems report definitions in a set of tables in an ORACLE database. Only one set of tables is needed to store report definitions in each ORACLE database with the transaction processing option. In ORACLE databases without the transaction processing option, each user must have his or her own set of tables.

RUNREP

The RUNREP program allows you to run pre-defined reports, sending the result to a browser (on most systems) or writing it directly to a file or printer. You can specify run-time parameters either on the command line or by filling in the Run-time Parameter Form (on most systems).

DUMPREP/LOADREP

The DUMPREP program creates an ASCII file containing report definitions. This file is portable and can be moved to other computers, including ones that use EBCDIC instead of ASCII. The LOADREP program loads the ASCII file into another ORACLE database, where reports can be edited or run.

GENREP

If you would like to use RUNREP to run a report that has not been executed before, you must first generate a file (called a runfile) containing a compressed version of the report definition. A runfile can be generated in any of the following ways:

- using either the Execute or Generate options of the Action Menu within SQLREP
- invoking GENREP to generate the file using report definitions stored in the database
- invoking GENREP using report definitions stored in an ASCII or EBCDIC file created by the DUMPREP utility.

PRINTDEF

With the PRINTDEF program, you can create your own printer definitions for displaying highlighted text and data, different fonts, etc.

TERMDEF

With the TERMDEF program, you can create your own terminal definitions for assigning different key mappings, etc. For more information, see Appendix G in the SQL*ReportWriter Reference Manual.

MOVEREP

MOVEREP is the SQL*ReportWriter Version 1.0 to 1.1 conversion program. For instructions on how to convert your reports, see “Upgrading from Prior Versions” in Appendix B of the SQL*ReportWriter Reference Manual.



PART

III

BUILDING REPORTS

REPORT FEATURES AND OUTPUTS

This chapter is divided into three sections:

- features listing
- report name listing
- visual (report output) index.

The features section contains an alphabetized listing of all report features (e.g., a control break) that are explained in depth in Chapter 4 or 5 of this book. Beside each feature, you will find one or more report names: those reports contain stepby-step instructions on how to build that feature.

The report name section contains an alphabetized listing of all reports in this book. Beside each name, you will find two page numbers: the first page number, the sample output page, is the page on which you will find a sample output and brief description of the report; the second page number, the explanation page, is the page on which you will find the detailed explanation of the report. The detailed explanation contains the concepts, organization, and building steps of the report.

The visual index section contains a sample report output of each report explained in this book. Below every sample output is a brief description, pointing to features that make the output unique from other outputs.

Features Listing

Below is an alphabetized list of all features that are explained in this book. To the right of each feature is the name of one or more report, found in Chapter 4 or 5, that contains the feature and explains how to build it. Once you find the name of the report that contains a feature you would like to build or learn more about, turn to page 3-4 to find the page on which the report is explained.

Feature	Report(s) with Feature Explained
&SQL	Matrix Ranking Report
Aggregating data	Aggregating Data Within Ranges Report
Boilerplate text	Form Letter Report
Changeable number of records per column	Changeable Number of Records per Column Report
Computed field	Computed Field Report
Computing running totals	Computations On Summaries: using cross-referenced queries, Running Totals-using SQL
Control break	Break Report, Computed Field Report, Intermixing Fields From Different Groups Across Break Report, Wrapped Break Report, Matrix Break Report
Column headings (deleting)	Suppressing Column Headers Report
Column headings (conditional)	Suppressing Column Headers Report
Conditional highlighting	Conditional Highlighting Report
Conditional printing	Conditional Printing Report
Crosstab print direction	Matrix Report
Default report	Tabular Report, Master/Detail Report
Display formats (altering)	Wrapped Break Report
Field labels (deleting)	Form Letter Report, Matrix Report
Field labels (modifying)	Form Letter Report, Summary Report, Time Series Report
First Function	Database Values In Page Headings Report
Footers (creating)	Group Footing Report
Mailing labels	Introductory Mailing Label Report, Advanced Mailing Labels Report
Last Function	Database Values In Page Headings Report
Matrix Groups (using)	Matrix Report, Matrix Ranking Report, Matrix Break Report

Feature	Report(s) with Feature Explained
Null queries	Relative Position Of Related Groups, Relative Position Of Unrelated Groups, Suppressing Column Headers Report
Page break of Always Parent/Child (Master/ Detail) relationships	Check Printing Report Master/Detail Report, Master/Detail/Summary Report
Print Group of Report	Subtotal Report
Print Group of Page	Placing Database Values In Page Headings Report
Print Group of a Group	Wrapped Break Report
PRT datatype	Creating Groups With No Printable Fields
Ranking data	Matrix Ranking Report, Ranking Report
Record spacing (modifying)	Master/Detail Report
Relative Position of fields	Introductory Mailing Label Report, Invoice Report
Relative Position of groups	Relative Position Of Related Groups, Relative Position Of Unrelated Groups,
Relative Position of Margin	Intermixing Fields From Different Groups, Check Printing Report
Reset Group of Report	Subtotal Report
Reset Group of a group	Wrapped Break Report
Skipped fields	Master/Detail Report, Master/Detail/Summary Report
Summary fields	Subtotal Report, Master/Detail/Summary Report
Totalling columns	Subtotal Report
Totalling rows	Computed Field Report
Unrelated queries	Master/Master Report
User exits	Conditional Highlighting Report, Conditional Printing Report, Invoice Report, Ranking Report
Variable-length fields	Form Letter Report
Word-wrapped fields	Advanced Mailing Label Report, Wrapped Break Report

Report Name Listing

Below is an alphabetized list of all reports that are explained in this book. Beside each name, you will find two page numbers: the sample output page, and the explanation page. On the sample output page, you will find a sample output and brief description of the report. On the explanation page, you will find a detailed explanation of the report, including the concepts, organization, and steps required to build the report. **Note:** Chapter 4 contains introductory reports; chapter 5 contains advanced reports.

Report Name	Sample Output Page	Explanation Page
Across Reports With Control Breaks Report	3-17	5-35
Advanced Mailing Report	3-35	5-50
Aggregating Data Within Ranges Report	3-16	5-28
Break Report	3-13	4-6
Changeable Number Of Records per Column Report	3-31	5-76
Check Printing Report	3-37	5-57
computations on summaries using cross-referenced queries	3-9	5-40
Computations On Summaries using SQL	3-10	5-44
Computed Fields Report	3-8	4-20
Conditional Highlighting	3-33	5-10
Conditional Printing	3-32	5-78
Creating Groups With No Printable Fields	3-30	5-6
Form Letter Report	3-36	4-24
Group Footing Report	3-27	4-46
Intermixing Fields From Different Groups: using the Margin setting	3-15	5-3
Introductory Mailing Label Report	3-34	4-16
Invoice Report	3-40	5-52
Master/Detail Report	3-19	4-38
Master/Detail/Summary Report	3-21	441
Master/Master Report	3-18	4-35
Matrix Ranking Report	3-22	4-34
Matrix Report	3-22	4-30
Matrix Report With Zeros For Null Values	3-24	5-63
Matrix Break Report	3-23	5-66
Page Heading Report	3-25	4-49
Placing Database Values In Page Headings Report	3-26	5-25

Report Name	Sample Output Page	Explanation Page
Printing Reports On Pre-printed Forms	3-39	5-30
Ranking Report	3-12	5-71
Relative Positioning Of Unrelated Groups Report	3-28	5-14
Relative Positioning Of Related Groups Report	3-29	5-18
Spelling Out Cash Amounts On Checks	3-38	5-60
Subtotal Report	3-7	4-11
Suppressing Column Headers When No Detail Records Are Retrieved	3-20	5-22
Tabular Report	3-6	4-3
Time Series Calculations Report	3-11	5-48
Wrapped Break Report	3-14	5-38

Visual Index

This section contains a sample output of each report explained in this book. Below every sample output, you will find a description of what makes the output that you are looking at different from other report outputs

Tabular Report

Deptno	Dname	Loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

A Tabular report is a “default” report. Information appears in a multi-column, multi-row format. Each column corresponds to a column in database table. SQL*ReportWriter uses the defaults to format and display the output. Column heading names (e.g., Deptno, Dname, etc.) are derived from the columns in your SELECT statement.

Level: Introductory. see chapter 4, page 3.

Subtotal Report (V1.0)

Deptno	Empno	Ename	Sal
10	7782	CLARK	2450
	7839	KING	5000
	7934	MILLER	1300
Total for Dept.			8750
20	7369	SMITH	800
	7876	ADAMS	1100
	7902	FORD	3000
	7788	SCOTT	3000
	7566	JONES	2975
Total for Dept.			10875
30	7499	ALLEN	1600
	7698	BLAKE	2850
	7654	MARTIN	1250
	7900	JAMES	950
	7844	TURNER	1500
	7521	WARD	1250
Total for Dept.			9400
Total for Company			29025

A Subtotal report contains a computed value printed below (by default) the column used to compute it. In the output above, the circled values are subtotals:

- each department's total is printed below the sal column because the department totals are computed by summing the Sal column for each department
- the company total is also printed below the Sal column, because it is computed by summing the Sal column for the entire report.

Level: Introductory. See Chapter 4, page 11.

Computed Fields Report (V1.0)

Sales Rep.	7499			
Custid	Dollars	Rank		Percent
-----	-----	-----	-----	-----
104	7160.8	1		90.98
107	710	2		9.02
Sales Rep.	7521			
Custid	Dollars	Rank		Percent
-----	-----	-----	-----	-----
106	9024.4	1		91.25
103	764	2		7.73
101	101.4	3		1.03
Sales Rep.	7654			
Custid	Dollars	Rank		Percent
-----	-----	-----	-----	-----
102	27775.5	1		100.00
Sales Rep.	7844			
Custid	Dollars	Rank		Percent
-----	-----	-----	-----	-----
108	6400	1		11.02
100	5280.9	2		9.10

A Computed Fields report contains at least one field, displayed in a column, whose values are determined by performing a calculation on another field. In this report, there are two computed fields RANK, and PERCENT. The RANK field has a function that numbers the records fetched for the DOLLARS field. The PERCENT field is calculated by dividing each value of the DOLLARS field by the total DOLLARS for that sales rep.

Level: Introductory. See Chapter 4, page 20.

Computations On Summaries using cross-referenced queries (V1.1)

Ordid	Prodid	Qty	Stdprice	Actprice	Totamt	Itemtot	Pctdisc
606	102130	1	3.4	3.4	3.4	3.4	0
	Sum				3.4	3.4	0
609	100861	1	45	35	45	35	22.22
	100870	5	2.8	2.5	14	12.5	10.71
	100890	1	58	50	58	50	13.79
	Sum				117	97.5	16.7
620	100860	10	35	35	350	350	0
	200376	1000	2.4	2.4	2400	2400	0
	102130	500	3.4	3.4	1700	1700	0
	Sum				4450	4450	0

A Computations On Summaries report typically comprises several directly computed fields, as well as computed fields which are based on other computed fields. An example of the latter is shown above. The circled value is a "computation of other computations" First, the TOTAMT and ITEMTOT fields are calculated for each PRODID. Then, the TOTAMT and ITEMTOT fields are summed for each ORDID. Finally the circled PCTDISC value, which is based on these summations, is calculated.

Level: Advanced. See Chapter 5, page 40.

Computations On Summaries using SQL (V1.0)

Ordid	Prodid	List Amount	Actual Amount	Cumulative List Amt	Cumulative Actual Amt	Cum Variance	Percent Ratio
601	200376	2.40	2.40	2.40	2.40	0.00	0%
602	100870	48.00	56.00	48.00	56.00	8.00	17%
		56.00	56.00	104.00	112.00	8.00	8%
603	100860	120.00	224.00	120.00	224.00	104.00	87%
		128.00	224.00	248.00	448.00	200.00	81%
		140.00	224.00	388.00	672.00	284.00	73%
604	100860	300.00	440.00	300.00	440.00	140.00	47%
		320.00	440.00	620.00	880.00	260.00	42%
		350.00	440.00	970.00	1,320.00	350.00	36%
	100861	78.00	84.00	78.00	84.00	6.00	8%
		84.00	84.00	162.00	168.00	6.00	4%
		90.00	84.00	252.00	252.00	0.00	0%
	100890	162.00	174.00	414.00	426.00	12.00	3%
		174.00	174.00	588.00	600.00	12.00	2%
605	100861	3,900.00	4,500.00	3,900.00	4,500.00	600.00	15%
		4,200.00	4,500.00	8,100.00	9,000.00	900.00	11%

A Computations On Summaries report typically comprises several “directly” computed fields, as well as computed fields which are based on other computed fields. On the previous page, an example of this type of report was described. This report is another example of computed fields which are based on other computed fields, but is constructed using SQL, and not SQL*ReportWriter features. 'Ibis report was included in this book to illustrate alternative techniques to building Computations On Summaries reports.

Level: Advanced. See Chapter 5, page 44.

Time Series Calculations Report (V1.0)

Custid	Shipdate	TOTAL	-Month Moving Average
100	30-JUL-86	\$3.40	\$3.40
	15-AUG-86	\$97.50	\$50.50
	01-JAN-87	\$730.00	\$730.00
	12-MAR-87	\$4,450.00	\$2,590.00
101	08-JAN-87	\$101.40	\$101.40
102	05-JUN-86	\$224.00	\$224.00
	20-JUN-86	\$56.00	\$140.00
	11-JAN-87	\$45.00	\$45.00
	05-FEB-87	\$23,940.00	\$11,992.50
	06-MAR-87	\$3,510.50	\$9,165.20
103	10-FEB-87	\$764.00	\$764.00
104	18-JUL-86	\$5.60	\$5.60
	25-JUL-86	\$35.20	\$20.40
	20-JAN-87	\$5,860.00	\$5,860.00
	04-FEB-87	\$1,260.00	\$3,560.00
105	03-MAR-87	\$46,370.00	\$46,370.00
106	30-MAY-86	\$2.40	\$2.40

Time series calculations involve averaging values over a specified period of time. In this example, a four-month average of purchases for each customer is calculated and displayed. The techniques described can be used to produce other formats of time series calculations as well.

Level: Advanced. See Chapter 5, page 48.

Ranking Report (V1.1)

Top 3 Customers:

Customer Name	Total Purchases
K + T SPORTS	\$46,370.00
VOLLYRITE	\$27,775.50
SHAPE UP	\$9,024.40

Top 75% of Sales:

Customer Name	Total Purchases
K + T SPORTS	\$46,370.00
VOLLYRITE	\$27,775.50

This report ranks data in two different ways by count, and by percentage. The upper Section displays the names and total purchases of the top three customers; the lower section displays the names and total purchases of those customers who constitute 75% of all sales. The ranking criteria can be set by the user at runtime, or default to previously specified values.

Level: Advanced. See Chapter 5, page 71.

Break Report (V1.0)

Deptno	Empno	Ename	Sal
10	7782	CLARK	2450
	7839	KING	5000
	7934	MILLER	1300
20	7369	SMITH	800
	7876	ADAMS	1100
	7902	FORD	3000
	7788	SCOTT	3000
	7566	JONES	2975
30	7499	ALLEN	1600
	7698	BLAKE	2850
	7654	MARTIN	1250
	7900	JAMES	950
	7844	TURNER	1500
	7521	WARD	1250

Break reports divide rows of a table into “sets” based on a common value in one of the columns. In the example shown above, all employees in department 10 are printed first, then those in department 20, etc. Furthermore, the department number is printed only once per set as shown in the circled values above.

Level: Introductory. See Chapter 4, page 6.

Wrapped Break Report (V1.1)

Name	Total	Pet. Total	Customers
EVERY MOUNTAIN	5860.00		5.66%
	1260.00		1.22%
	5.60		0.01%
	35.20		0.03%
	-----		-----
Sum	7160.80		06.91%
K + T SPORTS	46370.00		44.76%
	-----		-----
Sum	46370.00		44.76%
NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER	6400.00		6.18%
	-----		-----
Sum	6400.00		06.18%

This report is similar to the Break report shown on the previous page. The difference is that the break field is "wrapped" on word boundaries if it is too long to fit on one line.

Level: Advanced. See Chapter 5, page 38.

Intermixing Fields From Different Groups: using the Margin setting (V1.1)

Ename -----	(Dname -----	Sal -----
CLARK	ACCOUNTING	2450
KING		5000
MILLER		1300
Ename -----	Dname -----	Sal -----
ADAMS	RESEARCH	1100
FORD		3000
JONES		2975
SCOTT		3000
SMITH		800

Normally, a break field appears to the left of fields that are related to it (see the Break report described earlier). In this example, the break field appears between its related fields.

Level: Advanced. See Chapter 5, page 3.

Aggregating Data Within Ranges Report (V1.0)

Salary Range	Ename	Deptno
0 - 999	SMITH	20
	JAMES	30
1000 - 1999	ADAMS	20
	WARD	30
	MARTIN	30
	MILLER	10
	TURNER	30
	ALLEN	30
2000 - 2999	CLARK	10
	BLAKE	30
	JONES	20

Values from the database may be retrieved and formatted based on any aggregate range that you define. In this example report, the aggregate range is increments of one thousand, starting at zero.

Level: Advanced. See Chapter 5, page 28.

Across Reports With Control Breaks (V1.0)

Deptno	10					
Dname	ACCOUNTING					
Loc	NEW YORK					
Ename	CLARK	KING	MILLER			
	20					
	RESEARCH					
	DALLAS					
	ADAMs	FORD	JONES	SCOTT	SMITH	
	30					
	SALES					
	CHICAGO					
	ALLEN	BLAKE	JAMES	MART IN	TURNER	

An Across report prints database values of a column across the page instead of down (see the Ename values in the report above). Across reports are different from Down reports in that the x and y coordinates are reversed.

In Across reports with breaks, the master (or break) group prints "top to bottom," i.e., as it would in other master/detail reports. However, the values in the detail group prints across the page, instead of top to bottom. Thus, while DEPTNO, DNAME, and LOC print top to bottom, ENAME values print left to right in the example shown above.

Level: Advanced. See Chapter 5, page 35.

Master/Master Report (V1.0)

```
Total number of orders: 21  
Total number of customers: 9
```

This report displays aggregated data only, instead of individual values stored in the database. Each aggregate displayed is “independent,” that is, it is based on values in the database that have no direct relationship to other aggregates in the report.

Level: Introductory. See Chapter 4, page 35.

Master/Detail Report (V1.0)

Ename	Empno	Prodname	Amount	Custname		
ALLEN	7499	ACE TENNIS RACKET I	3000	EVERY MOUNTAIN		
		ACE TENNIS RACKET II	810	EVERY MOUNTAIN		
		ACE TENNIS BALLS-6 P	846.8	EVERY MOUNTAIN		
		SP TENNIS RACKET	24	EVERY MOUNTAIN		
		SP JUNIOR RACKET	1500	EVERY MOUNTAIN		
		RH : "GUIDE TO TENNIS	340	EVERY MOUNTAIN		
		SB ENERGY BAR-6 PACK	240	EVERY MOUNTAIN		
		SB VITA SNACK-6 PACK	400	EVERY MOUNTAIN		
		ACE TENNIS RACKET II	180	WOMENS SPORTS		
		ACE TENNIS BALLS-3 P	280	WOMENS SPORTS		
		ACE TENNIS BALLS-6 P	250	WOMENS SPORTS		
		MARTIN	7654	ACE TENNIS RACKET I	16569	VOLLYRITE
				ACE TENNIS RACKET II	2300.5	VOLLYRITE
ACE TENNIS BALLS-3 P	3306			VOLLYRITE		
ACE TENNIS BALLS-6 P	5600			VOLLYRITE		

Master/Detail reports contain two or more groups of data, and for every value of one group, the related values of the other group are fetched. For example, for every employee number and name of the first group, the employee's related information from the other group (PRODNAME, AMOUNT, CUSTNAME) is printed.

Level: Introductory. See Chapter 4, page 38.

Suppressing Column Headers When No Detail Records Are Retrieved (V1.0)

Deptno 20		
Dname RESEARCH	Ename	Job
Loc DALLAS	-----	-----
	SMITH	CLERK
	JONES	MANAGER
	SCOTT	ANALYST
	ADAMS	CLERK
	FORD	ANALYST
Deptno 30		
Dname SALES	Ename	Job
Loc CHICAGO	-----	-----
	ALLEN	SALESMAN
	WARD	SALESMAN
	MARTIN	SALESMAN
	BLAKE	MANAGER
	TURNER	SALESMAN
	JAMES	CLERK
Deptno 40		
Dname OPERATIONS		
Loc BOSTON		

In Master/Detail reports, the column headers appear for the detail group, even if no detail records are retrieved. This report shows how to suppress printing of these column headers when there are no associated details. For example, the column headers (EMPNO and JOB) of the detail group are not displayed because there are no employees in department 40.

The techniques described in this report can also be used to suppress detail group summaries when no records are retrieved for the detail group.

Level: Advanced. See Chapter 5, page 22.

Master/Detail/Summary Report (V1.0)

Product	Itemtot	Orderdate
ACE TENNIS BALL S-6 PAC	\$5.60	18-JUL-86
	\$11.20	25-JUL-86
	\$550.00	15-JAN-87
	\$280.00	22-FEB-87
ACE TENNIS RACKET I	\$3,000.00	15-JAN-87
ACE TENNIS RACKET II	\$810.00	15-JAN-87
RH : "GUIDE TO TENNIS"	\$340.00	22-FEB-87
SB ENERGY BAR-6 PACK	\$240.00	22-FEB-87
SB VITA SNACK-6 PACK	\$400.00	22-FEB-87
SP JUNIOR RACKET	\$1,500.00	15-JAN-87
SP TENNIS RACKET	\$24.00	25-JUL-86

Product Totals	Total
ACE TENNIS BALLS-6 PAC	\$846.80
ACE TENNIS RACKET I	\$3,000.00
ACE TENNIS RACKET II	\$810.00

A Master/Detail/Summary report is a Master/Detail report that contains one or more summaries. In this example, the top left rectangle is the master. The top right rectangle is the detail, and the bottom right rectangle is the summary. The summary adds the detail's item totals for each master record displayed.

Level: Introductory. See Chapter 4, page 41.

Matrix Report (V1.0)

	ANALYST	CLERK	MANAGER	PRESIDENT	SALESMAN	
10		1300	2450	5000	"	8750
20	6000	1900	2975			10875
30		950	2850		5600	9400
	[6000	4150	8275	5000	5600	29025

A Matrix report is like a grid: it contains one row of labels, one column of labels, and information in a grid format that is related to both the row and column "labels." (Matrix reports are also referred to as "Crosstab" reports.) This example report also contains three summaries one sums the salaries by department (8750. . .), one sums the salaries by job (6000.. .), one sums all salaries in the matrix (29025).

Matrix reports are different from Tabular reports in that the number of columns are not known in advance. Thus, this report would automatically be extended if a new job function, called RECEPTIONIST, was added to the underlying data tables.

Matrix reports can also be enhanced to rank the data. For example, the total salaries for each job could be ranked across the departments, putting a scored number beside each job's total salary.

Level: Introductory. See Chapter 4, page 30.

Matrix Break Report (V1.0)

Customer:	106						Total Purchases:	\$ 9,024
SHAPE UP								
Product	APR 86	MAY 86	JUN 86	JUL 86	AUG 86	SEP 86		
100860			440					
100861			84	4,500				
100870				1,400				
100890			174	290				
101860				1,200				
101863				900				
102130				34				
200376		2						
Customer:	106						Total Purchases:	\$ 280
VOLLYRITE								
Product	APR 86	MAY 86	JUN 86	JUL 86	AUG 86	SEP 86		

A Matrix Break report is a combination of Matrix and Break report formats. Essentially, a matrix-format report is printed for each master group value. This report shows specific information about each customer and their associated products.

Level: Advanced. See Chapter 5, page 66.

Matrix With Zeros For Null Values (V1.0)

	ANALYST	CLERK	MANAGER	PRESIDENT	SALESMAN
10	0	1	1	1	0
20	2	2	1	0	0
30	0	1	1	0	4

By default, matrix reports display a space for null (blank) values. In this example report, zeros replace null values.

Level: Advanced. See Chapter 5, page 63.

Page Heading Report (V1.0)

---- EMPLOYEE SUMMARY ----			
Deptno	Empno	Ename	Sal
10	7782	CLARK	2450
10	7839	KING	5000
10	7934	MILLER	1300
20	7369	SMITH	800
20	7566	JONES	2975
20	7788	SCOTT	3000
20	7876	ADAMS	1100
20	7902	FORD	3000
30	7499	ALLEN	1600
30	7521	WARD	1250
30	7654	MARTIN	1250
30	7698	BLAKE	2850
30	7844	TURNER	1500
30	7900	JAMES	950

A Page Heading report contains text and/or fields in the report's page header. The heading can span several lines. This example report is a Tabular report with boilerplate text (any text you enter) in the page header.

For an example of a report that has a database field in its page header, see the next report: Placing Database Values in Page Headers.

Level: Introductory. See Chapter 4, page 49.

Placing Database Values In Page Headings (V1.1)

Departments found on this page: 30 through 30																							
<table border="1"> <thead> <tr> <th colspan="4">Departments found on this page: 20 through 30</th> </tr> </thead> <tbody> <tr> <td colspan="4">Sal</td> </tr> <tr> <td colspan="4">-----</td> </tr> <tr> <td colspan="4">1500</td> </tr> <tr> <td colspan="4">1250</td> </tr> </tbody> </table>				Departments found on this page: 20 through 30				Sal				-----				1500				1250			
Departments found on this page: 20 through 30																							
Sal																							

1500																							
1250																							
Sal																							
3000																							
800																							
1600																							
2850																							
950																							
1250																							

Departments found on this page: 10 through 20			
Deptno	Ename	Job	Sal
-----	-----	-----	-----
10	CLARK	MANAGER	2450
	KING	PRESIDENT	5000
	MILLER	CLERK	1300
20	ADAMS	CLERK	1100
	FORD	ANALYST	3000
	JONES	MANAGER	2975

One or more database values can be located in the report's header or page header. In this example report, the first and last department number found on each page is displayed in the page header.

Level: Advanced. See Chapter 5, page 25.

Group Footing Report (V1.0)

Deptno	Empno	Ename	Sal
10	7782	CLARK	2450
	7839	KING	5000
	7934	MILLER	1300
----- End of Dept. 10 -----			
20	7369	SMITH	800
	7566	JONES	2975
	7788	SCOTT	3000
	7876	ADAMS	1100
	7902	FORD	3000
----- End Of Dept. 20 -----			
30	7499	ALLEN	1600
	7521	WARD	1250
	7654	MARTIN	1250
	7698	BLAKE	2850
	7844	TURNER	1500
	7900	JAMES	950

Group Footing reports contain boilerplate text, and/or fields, printed at the end of an instance of a group. In this sample output, the footer of the Employee group, "-----End of Dept.fieldvalue-----," is printed just before anew value is printed for the Department group (a new department number).

Level: Introductory. See Chapter 4, page 46.

Relative Positioning Of Unrelated Groups using null queries (V1.0)

Name	Job
EVERY MOUNTAIN	ANALYST
JOCKSPORTS	CLERK
JUST TENNIS	MANAGER
K + T SPORTS	PRESIDENT
NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER	SALESMAN
SHAPE UP	
TKB SPORT SHOP	Dname
VOLLYRITE	
WOWENS SPORTS	ACCOUNTING
	OPERATIONS
	RESEARCH
	SALES
Loc	
BOSTON	
CHICAGO	
DALLAS	
NEW YORK	

In SQL*ReportWriter, groups may be positioned to the Right, or Below other groups. This report shows how you can simulate a Relative Position of "left". This technique enables you to print the Job group to the right of the Name group, Dname group below the Job group, and Loc group to the left of Dname. This report is formatted correctly, regardless of the number of rows fetched for each group.

Level: Advanced. See Chapter 5, page 14.

Relative Positioning Of Related Groups: using null queries (V1.0)

Ordid	Orderdate	Shipdate	Custid	Name	Rep id						
610	07-JAN-87	08-JAN-87	101	TKB SPORT SHOP	7521						
<table border="0"> <thead> <tr> <th>Prodid</th> <th></th> </tr> </thead> <tbody> <tr> <td>100860</td> <td>610</td> </tr> <tr> <td>100870</td> <td>610</td> </tr> </tbody> </table>						Prodid		100860	610	100870	610
Prodid											
100860	610										
100870	610										
Ordid	Orderdate	Shipdate	Custid	Name	Rep id						
611	11-JAN-87	11-JAN-87	102	VOLLYRITE	7654						
<table border="0"> <thead> <tr> <th>Prodid</th> <th>Ordid</th> </tr> </thead> <tbody> <tr> <td>100861</td> <td>611</td> </tr> </tbody> </table>						Prodid	Ordid	100861	611		
Prodid	Ordid										
100861	611										

This report is similar to the one shown on the previous page, except that the groups are related. The groups in the lighter rounded rectangles are children of the group in the darker rounded rectangle.

Level: Advanced. See Chapter 5, page 18.

Creating Groups With No Printable Fields (V1.1)

Empno	Ename	Hiredate	Sal
7788	SCOTT	09-DEC-82	3000
7902	FORD	03-DEC-81	3000
7934	MILLER	23-JAN-82	1300
7369	SMITH	17-DEC-80	800
7876	ADAMS	12-JAN-83	1100
7900	JAMES	03-DEC-81	950
7782	CLARK	09-JUN-81	2450
7566	JONES	02-APR-81	2975
7698	BLAKE	01-MAY-81	2850
7839	KING	17-NOV-81	5000
7499	ALLEN	20-FEB-81	1600
7654	MARTIN	28-SEP-81	1250
7844	TURNER	08-SEP-81	1500
7521	WARD	22-FEB-81	1250

In most reports, you create groups and assign fields to the move or modify the appearance of fields in your report. Sometimes, however, you may find that you want to create a group that does not display a field in the report output.

In this example report, there is a field that has been assigned a Field Width of 0 that is being "printed" in the exact position that the Ename column appears.

Level: Advanced. See Chapter 5, page 6.

Changeable Number Of Records per Column Report (V1.1)

Ename	Sal				
SMITH	800	BLAKE	2850	ADAMS	1100
ALLEN	1600	CLARK	2450	JAMES	950
WARD	1250	SCOTT	3000	FORD	3000
JONES	2975	KING	5000	MILLER	1300
MARTIN	1250	TURNER	1500		

This report is similar to a Down/Across or Across/Down report, except that at runtime a user-specified number is used to format the report. The number tells SQL*ReportWriter how many rows should appear in each column. In this example, the user entered a 5, specifying that 5 records should appear in each column.

Level: Advanced. See Chapter 5, page 76.

Conditional Printing (V1.1)

Deptno	Dname	Ename	Sal
10	ACCOUNTING	CLARK	2450
		KING	5000
		MILLER	1300
		Sum Salary:	8750
		Min Salary:	1300
20	RESEARCH	ADAMS	1100
		FORD	3000
		JONES	2975
		SCOTT	3000
		SMITH	800
		Min Salary:	800

Conditional Printing reports display database values and/or text information only when certain criteria are met. You use this type of report when you want to determine which database values or text objects should be displayed in the report. In this example, a text object that contains the salary sum computation is displayed for department 10 only.

The conditional printing technique can also be used to produce form letters with varying contents.

Level: Advanced. See Chapter 5, page 78.

Conditional Highlighting (V1.1)

Empno	Ename	Sal
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7566	JONES	2975
7654	MARTIN	1250
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7839	KING	5000
7844	TURNER	1500
7876	ADAMS	1100
7900	JAMES	950
7902	FORD	3000
7934	MILLER	1300

Conditional Highlighting reports highlight database values and/or text when certain criteria are met. You use this type of report when you want database values to determine what parts of the report should be highlighted. In this example, SQL*ReportWriter holds the names and salaries of all employees whose salaries are greater than 2000.

Level: Advanced. See Chapter 5, page 10.

Introductory Mailing Label Report (V1.0)

NORTH WOODS HEALTH AND FITNESS 98 LONE PINE WAY HIBBING	MN 55649	TKB SPORT SHOP 490 BOLI RD. REDWOOD CITY	CA 94061
K + T SPORTS 3476 EL PASEO SANTA CLARA	CA 91003	SHAPE UP 908 SEQUOIA PALO ALTO	CA 94301
EVERY MOUNTAIN 574 SURRY RD. CUPERTINO	CA 93301	VOLLYRITE 9722 HAMILTON BURLINGAME	CA 95133
WOMENS SPORTS VALCO VILLAGE SUNNYVALE	CA 93301	JOCKSPORTS 345 VIEWRIDGE BELMONT	CA 96711
JUST TENNIS HILLVIEW MALL BURLINGAME	CA 97544		

This report automatically prints mailing labels in multiple columns on each page. The labels can be printed in one or many columns, and can begin at any position. Long names (e.g., North Woods Health) and addresses will be truncated. You can make the labels print across the page and then down (North Woods, K+ T Sports, Every Mountain...), or down and then across (North Woods, TKB Sport Shop, K + T Sports).

Level: Introductory. See Chapter 4, page 16.

Advanced Mailing Label Report (V1.1)

JOCKSPORTS 345 VIEWRIDGE SUITE 400-G BELMONT, CA 96711	EVERY MOUNTAIN 574 SURRY RD. CUPERTINO, CA 93301
TKB SPORT SHOP 490 BOLI RD. REDWOOD CITY, CA 94061	K + T SPORTS 3476 EL PASEO SANTA CLARA, CA 91003
VOLLYRITE 9722 HAMILTON BURLINGAME, CA 95133	SHAPE UP 908 SEQUOIA PALO ALTO, CA 94301
JUST TENNIS HILLVIEW MALL BURLINGAME, CA 97544	WOMENS SPORTS VALCO VILLAGE , SUNNYVALE, CA 93301

This report is similar to the report on the previous page, except that it suppresses blank address lines. In the previous example, the report will print a blank line for any null values that are retrieved from the database. For example, if you typically have two lines for the street (one for the street number and name, and one for the building and suite number), but one of your customers does not have an address that requires the two lines, the null line will appear as a blank line in the mailing label. In this example Null values are compressed.

Level: Advanced. See Chapter 5, page 50.

Form Letter Report (two techniques one with V1.0, one with V1.1)

```
Name :          SMITH
Employee #:     7369
Department:    20
Job :          CLERK

In compliance with Federal law, we are
confirming your current salary of $ 800.
If this amount conflicts with your
records, please notify your manager
sometime this week.

Thank you.

The Audit Department
```

Form Letter reports contain database values embedded in boiler plate text. (Boiler plate text is any text that you enter on the Text Screen.) In this example, the salary is displayed in the middle of the letter; however, database values may be embedded anywhere in the text. You can word-wrap the letter, word-wrapping the database values as well as the text.

Although it is not illustrated in this example report, you can highlight the text (boldface, etc.) and also conditionally highlight database values. An example of conditionally highlighting is as follows: if you were designing a form letter that described transaction activities for a financial institution, you could to bold only those account withdrawals that caused an account to go below the minimum balance.

Level: Introductory. See Chapter 4, page 24.

Check Printing Report (V1.1)

○	SUMMIT SPORTING GOODS	40736	SUMMIT SPORTING GOODS	INVOICE	○
○	ANYTOWN, CA 12345	30-NOV-1990	123 MAIN STREET	Sales Order No 025162	○
○			ANYTOWN, CA 12345		○
○	PAY TO THE ORDER OF <u>VOLLYRITE</u> \$		Ordid_ Shipdate_ -----Total		○
○	<u>XXXX AND XX/100</u>		602 20-JUN-86 56.00		○
○	DOLLARS		603 05-JUN-86 224.00		○
○	001077:12207000.: AUTHORIZED SIGNATURE		611 11-JAN-87 45.00		○
○			614 05-FEB-87 23,940.00		○
○	SUMMIT SPORTING GOODS	40737	SUMMIT SPORTING GOODS	INVOICE	○
○	123 MAIN STREET	30-NOV-1990	123 MAIN STREET	Sales Order No 025163	○
○	ANYTOWN, CA 12345		ANYTOWN, CA 12345		○
○			Ordid_ Shipdate_ -----Total		○
○	PAY TO ORDER OF <u>CANCELLED</u> \$		618 06-MAR-87 3,510.50		○
○	<u>XXXX AND XX/100</u>				○
○	DOLLARS				○
○	001077:12207000.: AUTHORIZED SIGNATURE				○

This Check Printing report is built so that the check appears on the left, and its related stub on the right. It could, however, be built so that the check appears above the stub.

There is one interesting aspect to building any check printing report: canceling checks for stubs that require more than one "page." The company name is printed on the check for the first page of the stub, and "CANCELLED" is printed on all checks thereafter for stubs that are continued onto more than one "page."

Level: Advanced. See Chapter 5, page 57.

Spelling Out Cash Amounts On Checks V1.1)

Name EVERY MOUNTAIN SEVEN THOUSAND ONE HUNDRED SIXTY AND 80/1 00	Sum Total \$7,160.80
Name JOCKSPORTS FIVE THOUSAND TWO HUNDRED EIGHTY AND 90/100	Sum Total \$5,280.90
Name JUST TENNIS SEVEN HUNDRED SIXTY-FOUR AND 0/100	Sum Total \$764.00
Name K + T SPORTS FORTY-SIX THOUSAND THREE HUNDRED SEVENTY AND 0/100	Sum Total \$46,370.00
Name NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER SIX THOUSAND FOUR HUNDRED AND 0/100	Sum Total \$6,400.00
Name SHAPE UP NINE THOUSAND TWENTY-FOUR AND 40/100	Sum Total \$9,024.40
Name TKB SPORT SHOP ONE HUNDRED ONE AND 40/100	Sum Total \$101.40

There are various ways you can spell out the amount of a check. The method shown in this report uses the ORACLE DECODE function and a lookup table to perform the conversion. This report can spell out amounts from zero to 999,999.99

Level: Advanced. See Chapter5, page60.

Printing Reports On Pre-printed Forms (V1.1)

Summit Sporting Goods, Inc.
123 Main Street
Anytown, CA 12345

Sales Order No. 068942

Salesrep ALLEN
Customer EVERY MOUNTAIN
574 SURRY RD.
CUPERTINO CA 93301

Date	Code	Product	Quantity	Price	Amount
18-JUL-86	100871	ACE TENNIS BALLS-6 P	1	\$5.60	\$5.60
25-JUL-86	100871	ACE TENNIS BALLS-6 P	2	\$5.60	\$11.20
	101860	SP TENNIS RACKET	1	\$24.00	\$24.00
15-JAN-87	100860	ACE TENNIS RACKET I	100	\$30.00	\$3,000.00
	100861	ACE TENNIS RACKET II	20	\$40.50	\$810.00
	100871	ACE TENNIS BALLS-6 P	100	\$5.50	\$550.00
	101863	SP JUNIOR RACKET	150	\$10.00	\$1,500.00
22-FEB-87	100871	ACE TENNIS BALLS-6 P	50	\$5.60	\$280.00
	102130	RH: "GUIDE TO TENNIS	100	\$3.40	\$340.00
	200376	SB ENERGY BAR-6 PACK	100	\$2.40	\$240.00
	200380	SB VITA SNACK-6 PACK	100	\$4.00	\$400.00

Item totals: \$7,160.80

This example report illustrates report for formatting techniques for printing reports on pre-printed forms. The report outlines the design methodology for printing reports on a company sales order form, in this case a mock up of a form for a fictitious company named Summit Sporting Goods.

Level: Advanced. See Chapter 5, page 30.

Invoice Report (V1.1)

Summit Sporting Goods, Inc. 456 Central Ave. Bigtown, CA 67890			INVOICE
Sales rep <u>A L L E N</u>			Sales Order No. 025162
customer	<u>EVERY MOUNTAIN</u> 574 SURRY RD., CUPERTINO, CA 93301		
Date	Code	Product Description	Amount
18-JUL-86	100871	ACE TENNIS BALLS-6 P	5.60
25-JUL-86	101860	SP TENNIS RACKET	24.00
	100871	ACE TENNIS BALLS-6 P	11.20
15-JAN-87	100860	ACE TENNIS RACKET I	3,000.00
	100861	ACE TENNIS RACKET II	810.00
	100871	ACE TENNIS BALLS-6 P	550.00
	101863	SP JUNIOR RACKET	1,500.00
22-FEB-87	102130	RH: "GUIDE TO TENNIS	340.00
			Total \$6,240.80

Invoice reports have several distinguishing characteristics

- The top part of the invoice typically contains a customer name, address, and other related information. This information must all print on specific line and column positions on the pre-printed invoice form. The report named, "Printing Reports on Pre-Printed Forms" in this book describes how to print on specific line and column positions, even if the number of lines of data being retrieved may vary at runtime.
- The middle section in most invoices contains the line-item details, including billing amounts. The number of items being printed will vary from customer to customer, and may require the invoice to span multiple pages.
- The bottom part of invoice reports contain customer billing information. Typically, this is a page total on all but the last page. On the last page, the page total is replaced with the total billing amount for that customer.

Level: Advanced. See Chapter 5, page 52.

CHAPTER

4

INTRODUCTORY REPORTS

This chapter describes the distinguishing features, concepts, organization, and building steps for each of the reports listed below:

- Tabular Report
- Break Report
- Subtotal Report
- Introductory Mailing Label Report
- Computed Field Report
- Form Letter Report
- Matrix Report
- Master/Master Report
- Master/Detail Report
- Master/Detail/Summary Report
- Group Footing Report
- Page Heading Report.

How To Use This Chapter

You can use this chapter in one of “three ways:

- You can create all of the reports in this chapter. This is an excellent way to learn how to build many types of reports, and to learn how to use the SQL*ReportWriter settings.

If you have just finished with the Tutorial in Chapter 1 and were unfamiliar with SQL*ReportWriter beforehand, we suggest that you go through the first couple of reports in this chapter to get a feel of the report-building process.

- You can build any report in this chapter. Each report description contains step-by-step instructions for building each report. For example, if you want to build a matrix report, you can go to the section describing the matrix report and follow the instructions.
- You can refer to one or several reports that contain features you would like to use in a report that you are building.

Chapter 3 contains a list of all SQL*ReportWriter features used to build the reports in this book. It also contains a sample output and brief description of each report in this chapter.

How To Load Sample Reports

Several introductory reports are shipped with SQL*ReportWriter. This chapter explains those reports in detail by describing the distinguishing features, concepts, organization, and steps to build each report.

To see the reports shipped with SQL*ReportWriter, do the following:

1. Type `load rep example.rex` while you are in the SQL*ReportWriter demo directory. You have to load the latest reports only once.

Note: If you already have the introductory reports from SQL*ReportWriter Version 1.1.10 or earlier, you must drop those reports before loading the latest reports. To do this, type `drop_ex.com user name /password` while in the demo directory. (The command syntax for your operating system may vary. For more information, see your Installation and User's Guide.) Then type `loadrep example.rex`.

2. Enter your username and password on the security screen.
3. Type `sqlrep` and then press Return.
4. Enter your username and password on the security screen.
5. Type `A` and the Action menu will be displayed.
6. Type `o` (not a zero, an o), and then enter the name of the report you wish to see. For the reports in this chapter, the report names are stated in the Concepts section.

Tabular Report (V1.0)

Deptno	Dname	Loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Distinguishing Features

A Tabular report is a “default” report. Information appears in a multi-column, multi-row format. Each column corresponds to a column in database table. SQL*ReportWriter uses the defaults to format and display the output. Column heading names (e.g., Deptno) are derived from the columns in your SELECT statement.

Concepts

The only settings you need to enter are the name and text of the query. You don't need to change the default values for the other settings.

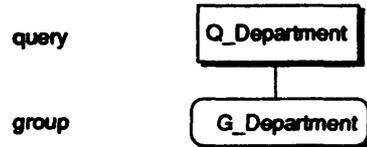
To see a sample Tabular report, open the SQL*ReportWriter example report named Tab_Report. See the section called “How To Load Sample Reports” on page 4-2 for details.

Organization

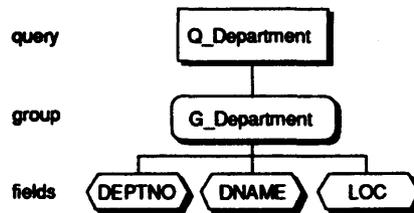
One query selects all of the fields that are displayed in this report. You'll name it Q_Department.

query Q_Department

There is one default group G_Department. SQL*ReportWriter automatically creates one default group for each query. The group name is simply the query name with a G_ prefixed to it. Therefore, if your query was named Accounts, the group name would be G_Accounts. (If your query has a Q prefix, SQL*ReportWriter removes the Q and replaces it with a G_.)



SQL*ReportWriter creates fields from the columns you select in your SELECT statement. The default field names are generated from the column name, appended with a table name if there is more than one field with the same name, or the alias if one is given. All fields in a tabular report belong to the default group, and the fields are displayed in the report output in the order in which you entered the columns in the SELECT statement.



Steps to Build this Report

Action Menu Each item on the Main Menu line (except Action) accesses a different set of SQL*ReportWriter screens. To choose a menu item, place the cursor on the item you want and press [Select], or press the first letter of the menu item you want; this both positions the cursor and selects the item.

To create a new report, follow these steps

1. Select Action from the Main Menu (press A).
2. Select New from the Action menu (press N).
3. Enter TabularReport and then press [Accept].

Query screen After you create anew report, you should specify the data that will be used init. To do so, name the new query (so that it can be referenced in the report) and then enter the SELECT statement.

4. Select Query from the Main Menu.
5. Enter `Q_DEPARTMENT` in the Query Name field. We named this query `Q_DEPAR'INIEN` because the SELECT statement will query information about all of the company's departments. You can, however, use any naming convention for your queries.
6. Press Next Field] to move the cursor to the SELECT Statement entry area.
7. Enter the following query:

```
SELECT DEPTNO, DNAME, LOC FROM DEPT
ORDER BY DEPTNO
```
8. Press [Accept] to return to the Main Menu.
9. Sekt Action from the Main Menu.
10. Select Execute from the Action Menu to execute the report.
11. To return to the Main Menu, press [Accept].

Break Report (V1.0)

Deptno	Empno	Ename	Sal
10	7782	CLARK	2450
	7839	KING	5000
	7934	MILLER	1300
20	7369	SMITH	800
	7876	ADAMS	1100
	7902	FORD	3000
	7788	SCOTT	30.00
	7566	JONES	2975
30	7499	ALLEN	1600
	7698	BLAKE	2850
	7654	MARTIN	1250
	7900	JAMES	950
	7844	TURNER	1500
	7521	WARD	1250

Distinguishing Features

Break reports divide rows of a table into "sets," based on a common value in one of the columns. In the example shown above, all employees in department 10 are printed first, then those in department 20, etc. Furthermore, the department number is printed only once per set as shown in the circled values above. For example, if you were to create a break report of all members of a family present at a company picnic, you would first write down the name of a family, and then next to their name you would write down all of the children in that family. You would then continue to the next family name, etc.

Smith	Dave JoAnne Jimmy
Anderson	Carol Peter

You use this type of report when you have one column repeating its value several times while columns related to that column change.

Concepts

A Break report is similar to a Master/Detail or Parent/Child report because one master record is printed (in this case, DEPTNO) and then all of the detail records with which the master has something in common are printed.

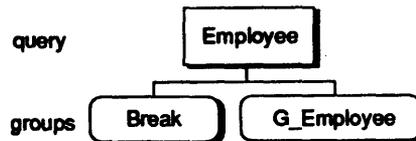
To see a sample Break report, open the SQL*ReportWriter example report named Break_Report. See the section called "How To Load Sample Reports" on page 4-2 for details.

Organization

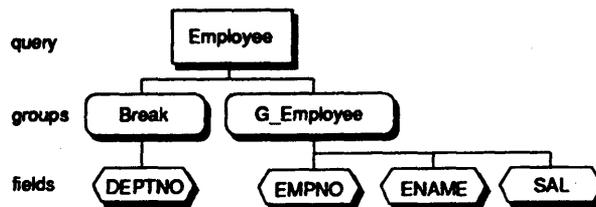
One query selects all of the fields that are displayed in the report.

query Employee

Two groups are owned by the query one group is created by default; one group is created manually and is assigned to the query. The default group is G_Employee, and the manually-created group is Break.



The break field, DEPTNO, is assigned to the manually-created group (Break). The rest of the fields, EMPNO, ENAME, and SAL, will remain assigned to the default group, G_Employee.



Steps to Build this Report

- Query Screen
1. Select Query from the Main Menu.
 2. Enter `EMPLOYEE` in the Query Name field.
 3. Press [Next Field] to move the cursor to the SELECT statement area and enter the following statement:

```
SELECT EMPNO, ENAME , DEPTNO, SAL FROM EMP
ORDER BY DEPTNO
```

4. Press [Accept] to return to the Main Menu.

- Group Screens
5. Select Group from the Main Menu.

Notice that there is a group named G_Employee. SQL*ReportWriter automatically creates one group for each query, using the query name and appending a G_ to name the group. You can also create groups. You need to do so to create a break report.

✓ Control Break

6. Press [Insert Record Above] while your cursor is on G_Employee. This will open up a line above G_Employee.
7. Enter `BREAK` in the Group Name field, and then press [Next Field] and enter `Q_Employee`. This tells SQL*ReportWriter that the new group, Break, will own one or more fields from the Q_Employee query. But at this time, SQL*ReportWriter does not know which data you want to belong to the group you just made.
8. Press [Scroll Right] to go to Group Screen Two, and then press [Next Field] until you reach the Record Spacing entry for Break. Enter a 1.
9. Press [Accept] to return to the Main Menu.

Field Screens

To assign data to the new group, you must assign the field(s) to the group.

10. Select Field from the Main Menu.
11. Press [Next Record] and [Next Field] until you reach the G_Employee group for the DEPTNO field.
12. Press [Delete Word] and then enter Break. This assigns the DEPTNO field to the group that you created, Break.

13. Press [Accept] to return to the Main Menu

Execute your report. Remember that when you are finished looking at your report, you return to SQL*ReportWriter by pressing [Accept].

Related Concepts

The break report concept can now be added to in two different ways. You can create two other types of break reports by:

- placing multiple fields in a break group
- having multiple break groups.

For example, when you place two fields, say DEPTNO and SAL, in a break group you get the following output. (To place SAL in the break group, do the following go to Field Screen One, move your cursor to the Group setting for the SAL field, delete G_Employee, enter Break, and execute your report. Delete the Record Spacing for the Break group, if desired.)

Deptno	Sal	Empno	Ename
10	2450	7782	CLARK
10	5000	7839	KING
10	1300	7934	MILLER
20	800	7369	SMITH
20	1100	7876	ADAMS
20	3000	7902	FORD
		7788	SCOTT
20	2975	7566	JONES
30	1600	7499	ALLEN
30	2850	7698	BLAKE
30	1250	7654	MARTIN
30	950	7900	JAMES
30	1500	7844	TURNER
30	1250	7521	WARD

DEPTNO prints more frequently than it did in the previous break report because all break group field values are printed unless the combination of those fields is repeated more than once. In this example, there are two people in department 20 that have a salary of 3000, so that combination of fields was only printed once.

When you create two break groups, and put one field in each break group, say DEPTNO and SAL, you get the following output. (To do this, you must go to Group Screen One, insert a group between Break and G_Employee, go to Field Screen One, move your cursor to the Group setting for the SAL field, delete the group name there, enter Break, and execute your report.)

<u>Deptno</u>	<u>Sal</u>	<u>Empno</u>	<u>Ename</u>
10	2450	7782	CLARK
	5000	7839	KING
	1300	7934	MILLER
20	800	7369	SMITH
	1100	7876	ADAMS
	3000	7902	FORD
		7788	SCOTT
	2975	7566	JONES
30	1600	7499	ALLEN
	2850	7698	BLAKE
	1250	7654	MARTIN
	950	7900	JAMES
	1500	7844	TURNER
	1250	7521	WARD

DEPTNO prints only once each time because there are several salaries for each department. SAL only prints once when there are more than one employee with the same salary in the same department.

Subtotal Report (V1.0)

Deptno	Empno	Ename	Sal
10	7782	CLARK	2450
	7839	KING	5000
	7934	MILLER	1300
Total for Dept.			8750
20	7369	SMITH	800
	7876	ADAMS	1100
	7902	FORD	3000
	7788	SCOTT	3000
	7566	JONES	2975
Total for Dept.			10875
30	7499	ALLEN	1600
	7698	BLAKE	2850
	7654	MARTIN	1250
	7900	JAMES	950
	7844	TURNER	1500
	7521	WARD	1250
Total for Dept.			9400
Total for Company			29025

Distinguishing Features

A Subtotal report contains a computed value printed below (by default) the column used to compute it. In the output above, the circled values are subtotals

- each department's total is printed below the Sal column because the department totals are computed by summing the Sal column for each department
- the company total is also printed below the Sal column, because it is computed by summing the Sal column for the entire report.

Concepts

In this example, you add two summary fields to the Break Report: one summary field prints the totals for each department; and the other summary field prints the grand total.

You create summaries on the Summary Screens. When you do so, you first enter a name for the field. Then, you choose the field that you wish to use in performing the calculation. That field will remain unchanged, and the calculated value(s) will be the value(s) for the summary field. Therefore, your report can display the output of both the field you used to perform the function and the summary field.

Next, you choose a function for the summary. Functions that are provided with SQL*ReportWriter are Sum, Min, Max, Count, Avg, %Total, Fint, Lust, and corresponding running functions. An example of a running function is R_Sum, the running sum.

Finally, you must assign a Print Group and then a Reset Group. The Print Group is the group where you want the summary to print-by default, the summary will appear in the footer of that group (for matrix reports the summary appears in the group's subfooter).

SQL*ReportWriter automatically assigns a logical Print Group, but you may change what is assigned by selecting from the List of values, deleting and entering or typing over a new group name. Note that you can also assign a Print Group of Report (you will do so for the grand total summary).

The Reset Group is the group at which you want to reset the summary accumulator's value to zero. You reset the summary so that a new value can be calculated. In this example (on the previous page), the total for each department is reset to zero just after each value is printed. In that way, the new field values for the next department can be summed without being added to the previous department's values. You can also assign a Print Group of Report (you will do so for the grand total summary).

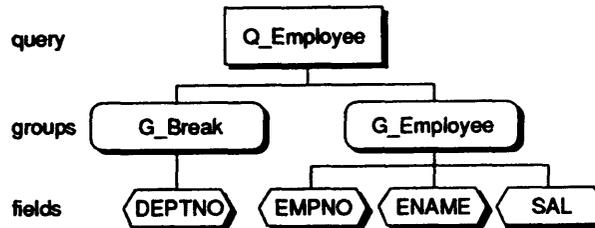
By default, the summary's value is placed directly below the last value of the field it summarizes. For that reason, the sum of each department's salaries appears below the Sal field. Also by default, the word "Sum" appears left-justified with the first field of the group owning the field. In this report, you will change that default text to Total for Dept. If the Print Group is Report, the word "Sum" appears left-justified with the first field of the report. In this report, you will change that default text to Total for Company.

To see a sample Subtotal report, open the SQL*ReportWriter example report named Sub_Report. See the section called "How To Load Sample Reports" on page 4-2 for details.

Organization

One query selects all of the fields printed in this report; two groups are assigned to that query. One group is created by default; one group is created and assigned manually to the same query.

In this example report, the break field, DEPTNO, is assigned to the manually created group, BREAK. The rest of the fields are assigned by default to the default group, G_Employee.



Steps to Build this Report

If you have already created the Break report, copy that report and name the copy Summary report. Then, skip down to the Summary Screens instructions, since the query, group, and field screens instructions are identical to those of the Break report.

Query Screen

Select Query from the Main Menu and enter `Q_EMPLOYEE` in the Query Name field.

Move the cursor to the SELECT Statement area and enter the following statement:

```
SELECT EMPNO, ENAME , DEPTNO , SAL FROM EMP  
ORDER BY DEPTNO
```

Press [Accept] to return to the Main Menu.

Group Screens

Select Group from the Main Menu.

✓ Control Break

Press [Insert Record Above] while your cursor is on G_Employee, and enter Break in the Group Name field. Press [Next Field] and enter Q_Employee. You must now assign the data that you want to belong to the group you just made.

Return to the Main Menu.

Field Screens

- 7. Select Field from the Main Menu and move your cursor until you reach the G_Employee group for the DEPTNO field.
- 8. Press [Delete Word] and then enter Break. This assigns the DEPTNO field to the group that you created, Break.
- 9. Return to the Main Menu.

Summary Screens

✓ Summary Fields

- 10. Select Summary from the Main Menu.
- 11. Enter the following values on the Summary Screen. (Remember, press [Next Field] or [Previous Field] to move left or right, and [Previous Record] or [Next Record] to move up or down.)

✓ Reset Group of Report

Summary Field	Field to Summarize	Function	Reset Group	Print Group
Sumsal	Sal	Sum	Break	Break
Grandsal	Sal	Sum	Report	Report

As you were entering the summary information, you may have noticed that some of the field values appeared automatically; for example, the Data Type. From the data you had entered, SQL*ReportWriter was able to determine those values for you and filled them in.

- 12. Return to the Main Menu.

Text Screen You will now modify the default summary label, Sum, to be a more descriptive field label.

- 13. Select Text from the Main Menu. The Text Screen is divided into two parts. The top half of this screen contains information about the text object (where it is to be positioned, how many blank lines to print before printing text, etc.), and the bottom half contains the actual text that will be printed when your report is executed.

✓ Field Labels (modifying)

14. Press [Next Record] until you reach the text screen with a text Object of Break and a Type of Footer. Move your cursor to the Text entry area by pressing [Next Field] several times. Then using your cursor keys, move to the S in Sum and type Total for Company over the word.
15. Now press [Next Record] until you reach the text Object of G_Employee and of Type Footer. Move to the Text entry area once again and change the word "sum" to Total for Dept..

Execute your report.

Introductory Mailing Label Report (V1.0)

NORTH WOODS HEALTH AND FITNESS 98 LONE PINE WAY HIBBING	MN 55649	TKB SPORT SHOP 490 BOLI RD. REDWOOD CITY	CA 94061
K + T SPORTS 3476 EL PASEO SANTA CLARA	CA 91003	SHAPE UP 908 SEQUOIA PALO ALTO	CA 94301
EVERY MOUNTAIN 574 SURRY RD. CUPERTINO	CA 93301	VOLLYRITE 9722 HAMILTON BURLINGAME	CA 95133
WOMENS SPORTS VALCO VILLAGE SUNNYVALE	CA 93301	JOCKSPORTS 345 VIEWRIDGE BELMONT	CA 96711
JUST TENNIS HILLVIEW MALL BURLINGAME	CA 97544		

Distinguishing Features

This report automatically prints mailing labels in multiple columns on each page. The labels can be printed in one or many columns, and can begin at any position. Long names (e.g., North Woods Health) and addresses will be truncated. You can make the labels print across the page and then down (North Woods, K + T Sports, Every Mountain..), or down and then across (North Woods, TKB Sport Shop, K + T Sports).

Concepts

Only positioning adjustments are made to this report after you enter the query name and SELECT statement. Set the Print Direction to Down/Across to allow the mailing labels to print down the page until it reaches the bottom. Then, the labels are printed in a column to the right of the first column of labels.

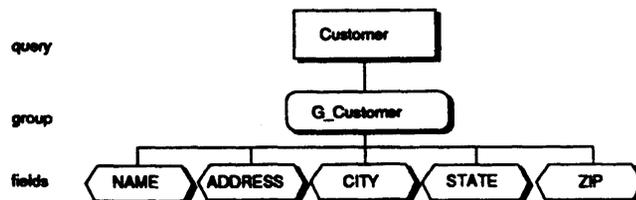
The Record Spacing entry on Group Screen Two controls the number of blank lines between each mailing label. The Spaces Before setting on Field Screen Two controls the inter-label spacing on a single row of labels. The width of each mailing label is controlled by the width of the largest address line. The fields are positioned so that they print in a rectangular format by setting the Relative Position of STREET and CITY to Below on Field Screen Two. As a result, the name appears on the first line, the street on the second line, and the city, state, and zip on the third line.

If there were two lines for the ADDRESS field (e.g., street on the first and then building number and suite on the second), and one of those values were null, this report would print a blank line for the missing value. In Chapter 5 there is an example report, Advanced Mailing Label Report, that removes null fields from each record before it prints them. Thus, no blank lines appear in the labels.

To see a sample Mailing Label report, open the SQL*ReportWriter example report named Mail_Label. See the section called "How To Load Sample Reports" on page 4-2 for details.

Organization

One query selects all of the fields for this report. There are no manually-created groups.



Steps to Build this Report

- Query Screen
1. Select Query from the Main Menu.
 2. Enter `customer` for the Query Name.

3. Press [Next Field] to move to the SELECT Statement area. You can enter queries in two ways by typing the statement, or by using the List of values to assist you. Use the List of values this time because there are several columns SELECTed from the same table. When you are through with this step, your SELECT statement will be:

```
SELECT NAME, ADDRESS, CITY, STATE, ZIP FROM CUSTOMER  
ORDER BY ZIP
```

Enter the word `SELECT`. Next, press [List]. A list of all tables you have access to appears. Move your cursor to the Customer table name. Press [List] again. A list of column names for the Customer table appears. Move your cursor to Name and press [select]. Move your cursor to Address and press [Select]. Do the same thing for the City, State, and Zip column names, and then press [Undo] to exit the list of column names. Select Customer from the table names list, and then press [Undo]. Now, enter the word, From, between zip and Customer and then type `ORDER BY ZIP`.

4. Press [Accept] to return to the Main Menu.

Group Screens

✓ Print Direction of
Down/Across

5. Select Group from the Main Menu.
6. Press [Next Field] until your cursor is in the Print Direction entry field and change the Print Direction of G_Customer to Down /Across. This causes one column of labels to be printed from the top to the bottom of the page, and then if there is room (in this case there is room) another column will print to the right of the first column, etc.

✓ Record Spacing
(modifying)

7. Press [Scroll Right] to move you to Group Screen Two, move your cursor to the Record Spacing entry field, and enter a 2. This puts two blank lines between each mailing label.
8. Press [Accept] to return to the Main Menu.

Field Screens

9. Select Field from the Main Menu and move your cursor to the Label entry field.
10. On Field Screen One, delete all of the Field Labels. (To do so, move to the Label entry area, press [Delete Word], then press [Next Record] and repeat until all field labels have been deleted.)
11. Move to Field Screen Two (press [Scroll Right]).

Change the Field Width of NAME to 30 , press [Next Record], change the Field Width of ADDRESS to 30 , and change the Field Width of CITY to 19 . This tells SQL*ReportWriter to truncate those fields to 30 (or 19) characters, including spaces.

✓ Relative Position
of fields

Move to the Relative Position entry field for ADDRESS and enter BELOW, and then enter BELOW for CITY, also. By default, SQL*ReportWriter places a next field to the right of a previous field. Changing the Relative Position of Address to Below tells SQL*ReportWriter to place the ADDRESS field below the NAME field. Likewise, the CITY field will be placed below ADDRESS. By not changing the default of the STATE and ZIP fields, SQL*ReportWriter will automatically place their values beside CITY.

Group Screens 14. Select Report from the Main Menu and change the Height to 24.
Execute your report.

Computed Field Report (V1.0)

Sales Rep.	7499			
Custid	Dollars	Rank	Percent	
-----	-----	-----	-----	
104	7160.8	1	90.98	
107	710	2	9.02	
Sales Rep.	7521			
Custid	Dollars	Rank	Percent	
-----	-----	-----	-----	
106	9024.4	1	91.25	
103	764	2	7.73	
101	101.4	3	1.03	
Sales Rep.	7654			
Custid	Dollars	Rank	Percent	
-----	-----	-----	-----	
102	27775.5	1	100.00	
Sales Rep.	7844			
Custid	Dollars	Rank	Percent	
-----	-----	-----	-----	
108	6400	1	11.02	
100	5280.9	2	9.10	

Distinguishing Features

A Computed Field report contains at least one field, displayed in a column, whose values are determined by performing a calculation on another field. In this report, there are two computed fields: RANK, and PERCENT. The RANK field has a function that numbers the records fetched for the DOLLARS field. The PERCENT field is calculated by dividing each value of the DOLLARS field by the total DOLLARS for that Sales Representative.

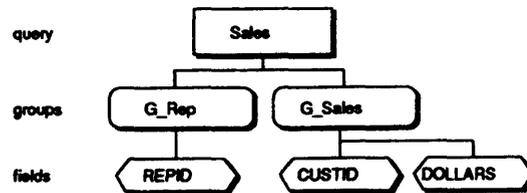
Concepts

Computed fields derive their value(s) from underlying database columns. To create a computed field, you must first create a new field on Field Screen One, specify a database column as the Source (the database column you want SQL*ReportWriter to perform a computation on), and specify a function (such as R_Count or percent). Lastly, you must specify a Reset Group on Field Screen Three, so that SQL*ReportWriter can correctly reset the computed field's accumulator during computation. In this example report, the accumulators for Rank and Percent are reset to zero just before the next Sales Representative's information is displayed. This prevents the Rank and Percent for the previous Sales Representative from being used to compute the next Sales Representative's Rank and Percent.

To see a sample Computed Fields report, open the SQL*ReportWriter example report named Comp_Report. See the section called "How To Load Sample Reports" on page 4-2 for details.

Organization

One query selects three fields for this report. The other two items that appear to be fields in the report output (RANK and PERCENT) are actually computed fields. The query owns two groups the default and one that is created manually (G_Rep). The presence of G_Rep causes a control break (see the Break Report for details).



Steps to Build this Report

- Query Screen
1. Select Query from the Main Menu.
 2. Enter `Sales` as the Query Name, and then enter the following SELECT statement:

```
SELECT REP ID, CUSTOMER . CUSTID, SUM (TOTAL) DOLLARS
FROM CUSTOMER, ORD
WHERE ORD.CUSTID = CUSTOMER. CUSTID
GROUP BY REP ID, CUSTOMER. CUSTID
ORDER BY REP ID, SUM (TOTAL) DESC
```

- | | |
|--|--|
| <p>Group Screens</p> <p>✓ Control Break</p> | <p>3. Select Group from the Main Menu.</p> <p>4. Create a group above the G_Sales group (press [Insert Record Above]). Name it G_Rep.</p> <p>5. Enter Sales for the Query of the G_Rep group. Assigning Sales to the G_Rep group tells SQL*ReportWriter that G_Rep will own one or more fields from the Sales query.</p> <p>6. Move to Group Screen Two and enter BELOW in the Relative Position entry field for the G_Sales group.</p> <p>7. Move to the Lines Before Entry field for G_Sales and enter a 1. This will put one blank line just before the fields that belong to the G_Sales group (CUSTID, DOLLARS, and later RANK, and PERCENT). On the same screen enter a 2 in the Record Spacing entry field for G_Rep.</p> <p>8. Move to Group Screen Three and enter Left in the Label Position entry field for G_Rep. This will place all field labels that belong to G_Rep (in this case Sales Rep.) to the left of its database value. You still need to assign a field to that group, so press [Accept] to return to the Main Menu.</p> |
| <p>Field Screens</p> <p>✓ Field Labels
(modifying)</p> <p>✓ Computed Field</p> | <p>9. Select Field from the Main Menu.</p> <p>10. Assign the group you just created, G_Rep, to the REPID field by deleting its default Group Name, G_Sales, and entering G_Rep.</p> <p>11. Move your cursor to the Label entry field for REPID and change the label from Repid to SALES REP. Field labels appear exactly as they are entered here: if you enter a label in all capital letters, it will be in all capital letters in the report. (Note: If the text panel that contains the field label has a Status of Edited, any changes to the Label entry field will not be reflected in the report unless the text object is deleted. Deleting the text object, going to the Main Menu, and then returning to the Text Screen causes SQL*ReportWriter to regenerate the default text.)</p> <p>12. Now onto creating computed fields Press [Insert Record Below] while your cursor is on the last field, DOLLARS.</p> |

Enter the following information on Field Screen One (the first field screen) and Field Screen Three (the last screen).

Field Screen One

Field Screen Three:

Field Name	Source	Group Name	Function	Reset Group
Rank	Dollars	G_sales	R_count	G_rep
Percent	Dollars	G_sales	%Total	G_rep

Change the Field Width of DOLLARS, RANK, and PERCENT to 10.
Execute your report.

Form Letter Report (two methods one with V1.0, one with V1.1)

Name: SMITH
Employee #: 7369
Department: 20
Job : CLERK

In compliance with Federal law, we are
confirming your current salary of \$ 800.
If this amount conflicts with your
records, please notify your manager
sometime this week.

Thank you.

The Audit Department

Distinguishing Features

Form Letter reports contain database values embedded in boilerplate text. (Boilerplate text is any text that you enter on the Text Screen.) In this example, the salary is displayed in the middle of the letter; however, database values may be embedded anywhere in the text. In "Method-Two" of the Steps To Build This Report, you word-wrap the letter, which word-wraps the database values as well as the text.

Although it is not illustrated in this example report, you can highlight the text (boldface, etc.), and also conditionally highlight database values. An example of conditionally highlighting is as follows if you are designing a form letter that described transaction activities for a financial institution, you can bold only those account withdrawals that caused an account to go below the minimum balance.

Concepts

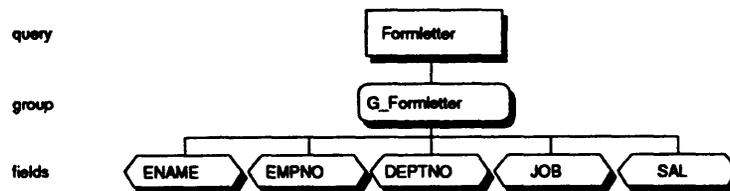
For this example report, you want to print the data for one employee on one page, mixing the data with boilerplate text (text you enter). To do this, you will modify the default report format and type in the text of the letter.

Selecting a Page Break of Always causes only one employee's data to be printed per page. You will also need to suppress the default field labels and will learn how to do so in two different ways. The field labels are suppressed by deleting them from the Column Heading of the Text Screen. Finally, editing the Body text Type causes the text to be displayed in the report output exactly as you entered it on the Text Screen.

To see a sample Form Letter report, open the SQL*ReportWriter example report named Form_Letter. See the section called "How To Load Sample Reports" on page 4-2 for details.

Organization

There is one query that selects all of the fields displayed in this report, and all of the fields belong to the default group.



Steps to Build this Report

There are two ways to build this report. The second method creates a preferable form letter output; however, build the first method to learn:

- about the Page Break setting
- about the List of values
- how fields and labels are referenced on the field screen
- how to delete field labels on the Text Screen
- how to create boilerplate text.

Method One:

- Query Screen
1. Select Query from the Main Menu and enter `Formletter` for the Query Name.
 2. Move to the SELECT Statement entry area (by pressing [Next Field]), and enter the following statement

```
SELECT ENAME, EMPNO, DEPTNO, JOB, SAL FROM EMP
```
 3. Press [Accept] to return to the Main Menu.
- Group Screens
4. Select Group from the Main Menu.
 5. Move to the Page Break entry field (by pressing [Next Field] a few times) and enter `Always`. This tells SQL*ReportWriter that each time it reads a new record from the database (e.g., if it had Smith's information and is now ready to read another employee's), SQL*ReportWriter should start a new page.
Note: Instead of typing the values, SQL*ReportWriter provides a List of values from which you may choose. Simply press [List], move your cursor to the value you wish to choose and press [Select]. A List of values is available whenever <List> appears at the bottom-right of your screen.
- Field Screens
6. Move to Field Screen Three and enter `Left` in the Align entry field for all of the fields but `SAL`.
Now that you have entered your query and set the Page Break to `Always`, all you need to do is modify the default text format and add your text.
- Text Screen
7. Select Text from the Main Menu, and press [Next Record] until you reach the text object that has `G_Formletter` as its Object, and Column Heading as its Type.
 8. Press [Next Field] until you reach the Text entry area.
 9. Press [Delete Line] until the default field labels are gone. This deletes the column headings in your report. (The suggested way to remove column headers is explained in Method Two. We included this method of deleting column headers so that you can learn about how text objects become Edited.)
- ✓ Page Break of Always
- ✓ Field Labels (deleting)

(Note: If you would have pressed [Delete Record], the column headings would have disappeared, also. However, pressing [Delete Record] causes the record to be returned to default text. For example, if you had entered or deleted any characters (including spaces) in the Text area, the Status would then become Edited. Try it. Add one or more characters. The Status will change once you leave the screen you are working on, so press [Next Record] and then [Previous Record]. Look in the top right-hand corner of the screen. The Status will now be Edited. Move to the Text area and press [Delete Record] to return the text Type to its default text. SQL*ReportWriter commits changes to the screen when you return to the Main Menu, so return to the Main Menu and then type T. If you return to that screen, the Status will now be Default The purpose of the Status indicator is described in "Method Two," later in this report. Press [Delete Line] until the default field labels are gone. This deletes the column headings in your report.)

10. Now press [Next Record] once. You will now see the field references (&ename, etc.). Field references have the same name as the field labels, but they are preceded with an ampersand (&).

Field references are SQL*ReportWriter's way to designate where the values from the database should go. If you delete a field reference, values for that field will not appear in the report. If you delete a label, the label will not appear. If you delete both, the field's label and values will not appear.

If you want a label to appear once, above all of its field values in the report, the label should appear in the Column Heading (this occurs by default). But if you want the label to appear each time a new record is read from the database (i.e., once for Smith, once for Jones, etc.) then the label should appear in the Body (the text Type should be Body). In this report, you want the labels to appear for each employee, so you will now add them to the body before their field reference.

✓ Boilerplate Text

11. Type in the field labels and text so that your screen looks exactly like the one below.

```
Name:           &ENAME
Emp. #:         &EMPNO
Department:    &DEPTNO
Job :          &JOB
```

```
In compliance with Federal Law, we
are confirming your current salary
of $ &sal. If this amount conflicts
with your records, please notify your
manager sometime this week.
```

```
Thank you.
```

```
The Audit Department
```

Execute your report.

The second way to produce this report makes the salary fit right in with the text, instead of having gaps appear around the embedded SAL field. To produce this report using the second method, you will continue to use the Formletter report.

To begin the second method, you first need to delete some of the edited text objects to regenerate their defaults. The most important thing to remember about the Text Screen is this: SQL*ReportWriter will only modify the text of a given Text Type (e.g., Column Heading) if the text Type does not have a Status of Edited.

Method Two:

- | | |
|---------------|---|
| Text Screen | 1. Move your cursor to the G_Formletter Column Heading and press [Delete Record]. |
| Field Screens | 2. Go to the Field Screens.
3. Move your cursor to the Label entry field and delete the labels for all of the fields (DEPTNO, EMPNO, etc.). Deleting labels is another way to prevent the field labels from appearing in your report. (Therefore, you will not need to press [Delete line] in the G_Formletter Column Heading on the Text Screen this time.) |

✓ Variable-length fields

4. Press [Scroll Right] twice to reach Field Screen Three, and enter Variable in the Align field entry for SAL.

Execute the report. Notice how the Variable alignment affected your report: it caused all of the field values to word-wrap. You would have gotten the same word-wrapped output that is on your screen right now had you set any or all of the field alignments to Variable. Thus, if you wish to word-wrap all fields that are found in the same text Object and Type, you need only set the Align of one of the fields to Variable.

Text Screen Although it may look like it will require many steps to get this report in the right format, you only need to do one more thing: add carriage (hard) returns to the text object.

✓ Carriage Returns (embedding)

5. Move to the Text Screen and press [Next Record] until you reach the G_Formletter Object and Body text Type. Move to the Text area and enter &CR at the end of the ENAME, EMPNO, DEPTNO, and JOB fields (leave one space between; e.g., &ENAME &CR). &CR causes the text that follows to go to the next line (it prevents word-wrapping). Your text should look like that below:

```
Name:           &ENAME &CR
Emp. #:         &EMPNO &CR
Department:     &DEPTNO &CR
Job :           &JOB &CR
&CR
In compliance with Federal Law, we
are confirming your current salary
of $ & sal. If this amount conflicts
with your records, please notify your
manager sometime this week. &CR
&CR
Thank you. &CR
&CR
The Audit Department
```

Execute your report.

Matrix Report (V1.0)

	ANALYST	CLERK	MANAGER	PRESIDENT	SALESMAN	
10		1300	2450	5000		8750
20	6000	1900	2975			10875
30		950	2850		5600	9400
	-----	-----	-----	-----	-----	-----
	<u>6000</u>	<u>4150</u>	<u>8275</u>	<u>5000</u>	<u>5600</u>	29025

Distinguishing Features

A Matrix report is like a grid: it contains one row of labels, one column of labels, and information in a grid format that is related to both the row and column 'labels.' (Matrix reports are also referred to as "Crosstab" reports.) This example report also contains three summaries: one sums the salaries by department (8750. . .), One sums the salaries by job (6000.. .), one sums all salaries in the matrix (29025).

Matrix reports are different from Tabular reports because the number of columns are not known in advance. Thus, this report would automatically be extended if a new job function, called RECEPTIONIST, was added to the underlying data tables.

Concepts

In the example report shown above, there are three default groups. One group prints the vertical labels, one group prints the horizontal labels, and the third group fills in the matrix. To create the matrix report, you must specify that the default groups are Matrix Groups, and then assign an appropriate Print Direction to them. In this example, the departments have a Print Direction of Down, the jobs have a Print Direction of Across, and the salaries have a Print Direction of Crosstab. Remember this You only use the Crosstab Print Direction in matrix reports.

This report also contains three summary fields a grand total, a sum of the salaries per department, and a sum of the salaries per job.

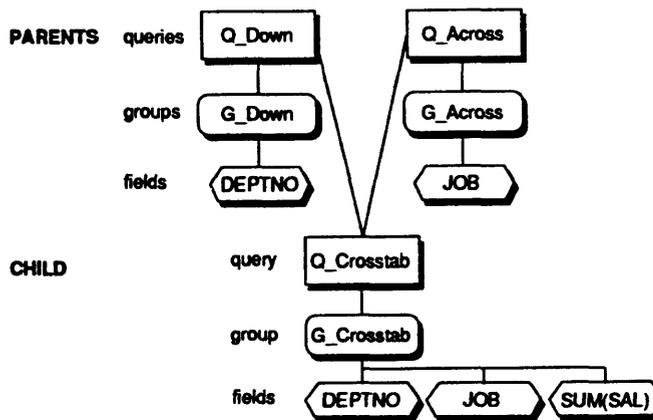
There are some rules that you should know when building matrix reports

- You must use exactly three queries, and they must be unique.
- Each query must have only one group.
- Each group must be specified as a Matrix Group.
- One group must have a Print Direction of *Across*, another of *Down*, and the third, *Crosstab*.

To see a sample Matrix report, open the SQL*ReportWriter example report named Matrix_Report. See the section called "How To Load Sample Reports" on page 4-2 for details.

Organization

This report contains three queries. Two of the queries, those that make the labels of the matrix, are parents of the third query, which fetches the crosstab information. The child query must be associated with both parent queries.



Three summary fields are also present in this report: one summary field is for the sum of salaries by department, another is for the sum of salaries by job, and the last is for the grand total.

Steps to Build this Report

Query Screen

1. Create a query with a Query Name of `Down` and a SELECT statement of

```
SELECT DISTINCT DEPTNO FROM EMP
```

2. Create a second query (press [Insert Record Below]) with a Query Name of `Across` and a SELECT statement of

```
SELECT DISTINCT JOB FROM EMP
```

3. Create a third query with a Query Name of `Crosstab` and a SELECT statement of

```
SELECT DEPTNO, JOB, SUM (SAL) FROM EMP
GROUP BY DEPTNO, JOB
ORDER BY DEPTNO, JOB
```

✓ Parent/Child Relationship"

and then press [Next Field]. This moves the cursor to the Parent Query 1 entry field. Enter `Down`. This tells SQL*ReportWriter that the `Down` query is a parent of the `Crosstab` query (or master of this master/master/detail relationship). Press [Next Field] again. This moves the cursor to the Parent Query 2 entry field. Enter `Across`. This tells SQL*ReportWriter that the `Across` query is also a Parent of the `Crosstab` query. You must now specify how the two parent queries are related to the child query, `Crosstab`. Press [Next Field] and then enter the following relationship information:

Child Columns	Parent 1 Columns	Parent 2 Columns.
Deptno	Deptno	
Job		Job

Note: You are not required to name your queries `Down`, `Across`, and `Crosstab`. We've done so in this example to help you understand how the Print Directions, discussed in step 4, affect your report output.

✓ Matrix Groups

Group Screens

Move to the Group Screens, enter an x in the Matrix Group setting for all three groups, and then enter the following Print Directions

Group Name	Matrix Group	Print Direction
Down	x	Down
Across	x	Across
Cross tab	x	Crosstab

- Field Screens
5. Move to Field Screens and delete all of the field labels. (To do this quickly, move to the Label entry field and press [Delete Line] and then [Next Record] repeatedly. This will delete the default labels so that they will not appear in the report.) The field values will serve as the “labels” of the matrix.
 6. Move to Field Screen Two and change the Field Width of SUM_SAL to 10. This will display the field’s database values in the 10-character width, truncating the data if it is more than 10 characters long. (If you don’t want to reduce the Field Width, because you do not want the field’s data to be truncated, you can increase the default report Width on the Report Screen.)
 7. Skip the JOB2 and DEPTNO2 fields by placing an x in their Skip entry fields. This will prevent those fields from being displayed in the report output. The only reason you put those fields in the Crosstab query was so that the Crosstab query could be pinned with its parent queries (in the Parent/Child Relationships entry area).

- Summary Screens
8. You could execute the report as it is, but let’s add a few summaries to make the report a little more informative. Move to the Summary Screens and add the following information.

Summary Screen One:

<i>Summary Name</i>	<i>Field to Summarize</i>	<i>Function</i>
sum_dept	SUM_SAL	Sum
sum_job	SUM_SAL	Sum
sum_total	SUM_SAL	Sum

Summary Screen Two:

<i>Print Group</i>	<i>Reset Group</i>
G_Down	G_Down
G_Across	G_Across
Report	Report

Execute the report. Notice that the Job fields (e.g., ANALYST) are not quite aligned with the salaries. This is because the default alignment for CHAR fields is Left and NUMBER fields is Right. You can change the default by changing the Align Setting of the JOB field to Right. So,

✓ Align Fields

9. Move to the Align entry field on Field Screen Three and enter the word `Right` for the JOB field.

Execute the report once again.

Related Concepts (V1.1)

You can enhance the basic matrix report by adding ranking to it. In the example below, the total salaries for each job are ranked. To add ranking to a matrix report, you create a new field whose Source is a SELECT statement. (See the SQL*ReportWriter Reference Manual for details.)

	ANALYST	CLERK	MANAGER	PRESIDENT
10		1300 2	2450 3	5000 1
20	6000 1	1900 1	2975 1	
30		950 3	2850 2	
	-----	-----	-----	-----
	6000	4150	8275	5000

To see a sample Matrix Ranking report, open the SQL*ReportWriter example report named Matrix_Ranking-Report. See the section called "How To Load Sample Reports" on page 4-2 for details.

✓ &SQL

- Field Screens 1. Move to Field Screen One and create a new field named RANK below SUM_SAL. Move to the Source column and enter the following

```
&SQL SELECT SUM (COUNT (DISTINCT DEPTNO) ) INTO : RANK FROM EMP
GROUP BY DEPTNO, JOB HAVING JOB = : JOB
AND SUM ( SAL) >= : SUM_ SAL
```

This statement counts each distinct department, computes the sum, and ranks the salaries for the same job in each department.

Assign RANK to the G_Crosstab group, and delete its Label. Do this by moving to the Label column and pressing [Delete Line].

Move to Field Screen Two and change the DataType of RANK to NUM. You must do this because a field whose Source uses &sql is automatically assigned a DataType of CHAR. Also, change RANK's Field Width to 1.

Move to Field Screen Three and the Align entry field for JOB and delete Right.

Execute your report. Notice that the cells are now ranked within each job.

Master/Master Report (V1.0)

```
Total number of orders: 21
Total number of customers: 9
```

Distinguishing Features

This report displays aggregated data only, instead of individual values stored in the database. Each aggregate displayed is "independent," that is, it is based on values in the database that have no direct relationship to other aggregates in the report.

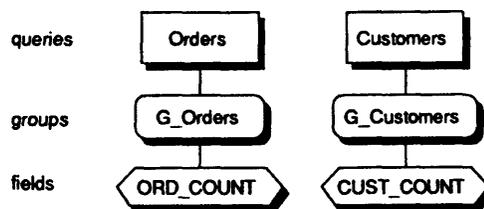
Concepts

A Master/Master report is also called a Parent/Parent report. It contains two queries that contain no parent/child relationships. In this example, each query counts the number of records it fetches; one query determines the total number of orders, the other the total number of customers. The calculation is performed by the SELECT statement.

To see a sample Master/Master report, open the SQL*ReportWriter example report named Master_Report. See the section called "How To Load Sample Reports" on page 4-2 for details.

Organization

A master/master report always contains two unrelated queries. In this example report, each query selects only one field, although they could select any number of fields. There are no manually created groups.



Steps to Build this Report

- Query Screen
1. Create a query with a Query Name of Orders and a SELECT statement of

```
SELECT COUNT (DISTINCT (ORDID) ) ORD_COUNT FROM ORD
```
 2. Create a query with a Query Name of Customers and a SELECT Statement of

```
SELECT COUNT (DISTINCT ( CUSTID ) ) CUST_COUNT FROM CUSTOMER
```

- Group Screens
3. Move to the Group Screens and press [Scroll Right] until you reach the Relative Position entry field. Enter Below for G_Customers.
 4. Now move to the Spaces Before entry field for G_Customers and enter a 0. This tells SQL*ReportWriter to place the group left-aligned with the G_Orders group. By default SQL*ReportWriter places a second parent group one space from the left margin of the first parent group.

✓ Field Labels (deleting)

- Field Screens
5. Now move to the Field Screens and delete both of the field labels. This prevents you from having to delete them from the Text Screen.

✓ Field Labels (modifying)

- Text Settings
6. Move your cursor to the Text Screen and press [Next Record] until you reach the screen where the Object is G_Orders and the Type is Body. Add the following text before the field reference (specified by &fieldname):

```
Total number of orders:
```
 7. Now do the same thing for the text Object of G_Orders and Type of Body. Add the following text:

```
Total number of customers:
```

Execute the report. Notice that there is a rather large gap between the colon and the number of orders/customers. This is because numbers are right-justified by default and the Field Width for those two fields (taken from the database width for that column) is much greater than the actual database value. So:

- Field Screens 8. Move to the Field Screens once more and press [Scroll Right] until you reach the Align entry field (on Field Screen Three). Enter `Left` for both the `ORD.COUNT` and `CUST_COUNT` fields.

Execute the report again to see the result.

Master/Detail Report (V1.0)

Ename	Empno	Prodname	Amount	Custname
ALLEN	7499	ACE TENNIS RACKET I	3000	EVERY MOUNTAIN
		ACE TENNIS RACKET II	810	EVERY MOUNTAIN
		ACE TENNIS BALLS-6 P	846.8	EVERY MOUNTAIN
		SP TENNIS RACKET	24	EVERY MOUNTAIN
		SP JUNIOR RACKET	1500	EVERY MOUNTAIN
		RH : "GUIDE TO TENNIS	340	EVERY MOUNTAIN
		SB ENERGY BAR-6 PACK	240	EVERY MOUNTAIN
		SB VITA SNACK-6 PACK	400	EVERY MOUNTAIN
		ACE TENNIS RACKET II	180	WOMENS SPORTS
		ACE TENNIS BALLS-3 P	280	WOMENS SPORTS
		ACE TENNIS BALLS-6 P	250	WOMENS SPORTS
MARTIN	7654	ACE TENNIS RACKET I	16569	VOLLYRITE
		ACE TENNIS RACKET II	2300.5	VOLLYRITE
		ACE TENNIS BALLS-3 P	3306	VOLLYRITE
		ACE TENNIS BALLS-6 P	5600	VOLLYRITE

Distinguishing Features

Master/Detail reports contain two or more groups of data, and for every value of one group, the related values of the other group are fetched. For example, for every employee number and name of the first group, the employee's related information from the other group (PRODNAME, AMOUNT, CUSTNAME) is printed.

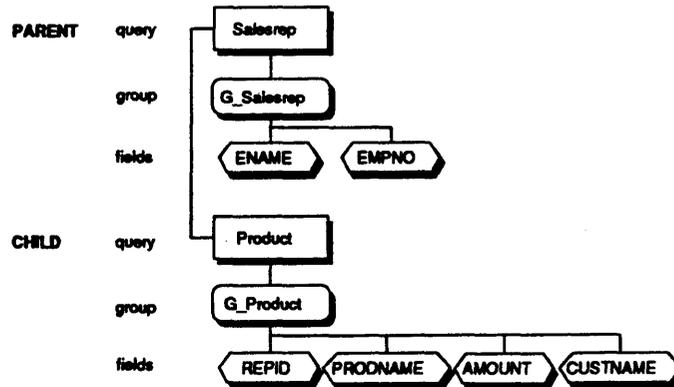
Concepts

There are two queries in a simple Master/Detail report, one of which is the parent of the other. (A Master/Detail report can also be called a Parent/Child report; however, the name, "Master/Detail report," is more common.) For each record retrieved for the parent query, related records of the child query are fetched. To join, or relate, the two queries, you specify parent/child relationships on the Query Screen.

To see a sample Master/Detail report, open the SQL*ReportWriter example report named Master_Detail_Report. See the section called "How To Load Sample Reports" on page 4-2 for details.

Organization

This report contains two queries, and there are no manually created groups. One of the queries, Salesrep, is the parent of the other, Product. While there may be more than two queries in a Master/Detail Report, there must be at least two queries, and one must be the parent of the other. There may be any number of groups in a Master/Detail Report.



Steps to Build this Report

✓ Parent/Child Relationships

- Query Screen 1. Create a query with a Query Name of `Salesrep` and a SELECT statement of

```
SELECT DISTINCT ENAME , EMPNO FROM EMP
WHERE JOB = 'SALESMAN'
```

2. Create another query (press [Insert Record Below]) with a Query Name of `Product` and a SELECT statement of

```
SELECT REP ID, PRODNAME, AMOUNT, CUSTNAME
FROM SALES
ORDER BY REP ID , CUSTNAME
```

and then press [Next Field] so that your cursor is in the Parent Query 1 entry field. Enter `Salesrep`. You want the Parent Query to be `Salesrep` because you want the sales representative's name and employee number to print once for all of the sales representative's product information retrieved.

You must now specify how the parent query is related to the child (Detail), Product. Press [Next Field] and then enter the following so that the bottom portion of your screen contains the information below:

Child Columns	Parent 1 Columns	Parent 2 Columns
Repid	Empno	(Blank)

Specifying REPID and EMPNO in the Matching Columns area is a way of specifying a pin condition. In this example, it is analogous to writing both queries as one query with a pin statement of

```
WHERE EMPNO = REP ID (+)
```

Whether you should use pins or multiple queries depends on the distribution of master and detail records. If there are many detail records per master, even if the master only has a few columns, a multiple query will most likely be faster. When you have more than two-way situations, especially when the ratio of parents to children is high, pins are more likely to be slower than multiple queries. (Multiple queries are also easier to explain and maintain than pins.)

Group Screens Move your cursor until you reach the Record Spacing entry field for G_Salesrep and enter a 1. This tells SQL*ReportWriter that you want it to print the first record of the G_Salesrep group (in this case Allen and 7499), then print the related detail records (the Prodnames, Amounts, and Custnames), then print one blank line, and then print the next record of the G_Salesrep group (in this case Martin and 7654).

Field Screens Move to the Field Screen Two and change the Field Widths for PRODNAME to 20, AMOUNT to 10, and CUSTNAME to 15.

✓ **Skipping Fields** Now move to Field Screen Three and Skip the REPID field (place an x in the Skip entry field for REPID).

Execute your report.

Master/Detail/Summary Report (V1.0)

Name	Product	Itemtot	Orderdate
EVERY MOUNTAIN	ACE TENNIS BALLS-6 PAC	\$5.60	18-JUL-86
		\$11.20	25-JUL-86
		\$550.00	15-JAN-87
		\$280.00	22-FEB-87
	ACE TENNIS RACKET I	\$3,000.00	15-JAN-87
	ACE TENNIS RACKET II	\$810.00	15-JAN-87
	RH : "GUIDE TO TENNIS"	\$340.00	22-FEB-87
	SB ENERGY BAR-6 PACK	\$240.00	22-FEB-87
	SB VITA SNACK-6 PACK	\$400.00	22-FEB-87
	SP JUNIOR RACKET	\$1,500.00	15-JAN-87
	SP TENNIS RACKET	\$24.00	25-JUL-86

Product Totals	Total
ACE TENNIS BALLS-6 PAC	\$846.80
ACE TENNIS RACKET I	\$3,000.00
ACE TENNIS RACKET II	\$810.00

Distinguishing Features

A Master/Detail/Summary report is a Master/Detail report that contains one or more summaries. In this example, the top left rectangle is the master. The top right rectangle is the detail, and the bottom right rectangle is the summary. The summary adds the details item totals for each master record displayed.

Concepts

This report contains four groups. The first group prints the customer name, the second group prints the name of the product ordered (this group is created so that the product name is not repeated every instance), the third group prints the date and cost of each product, and the fourth prints the summary data. Control breaks are established by both the customer group and the product group.

To create this Master/Detail/Summary report, you use three queries that specify data from four tables Ord, Item, Customer, and Product. Within the SELECT statements, you must query the columns needed for the report as well as the columns that pin the tables together.

Four groups are built from the data in the three queries. Three of the groups contain all the fields from their respective queries, and one group is manually created and contains the “Descrip” (i.e., product name) field. Two control breaks are made with these groups G_Cust breaks by customer, and G_Prod breaks byproduct.

The summary group (G_Summary) is very different from a simple break total. This group, easily created with a SQL statement, provides summary information broken down byproduct. As with all groups in SQL*ReportWriter, you can place this group wherever you want in your report.

In the Field Screens, you will add some Display Formats and Field Widths, and Skip the fields which were queried only to join the tables.

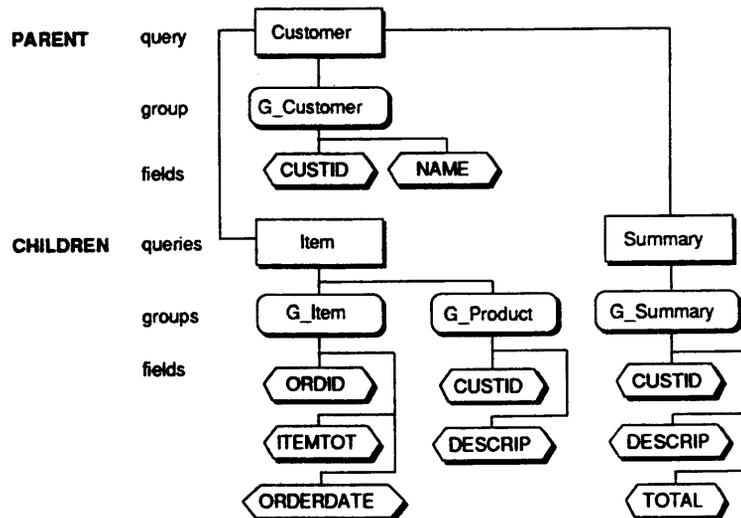
You do not use the Summary Screens because the summary is created with a SQL statement.

Boilerplate text is not added in this example, but you can manipulate the default format slightly to make the report look like the example.

To see a sample Master/Detail/Summary report, open the SQL*ReportWriter example report named Mast_Det_Sum_Report. See the section called “How To Load Sample Reports” on page 4-2 for details.

Organization

This report contains three queries; the Customer query is the parent of the Item and the Summary queries.



Steps to Build this Report

Query Screens

1. Create a query with a Query Name of *customer* and a SELECT statement of

```
SELECT CUSTID, NAME FROM CUSTOMER
ORDER BY NAME
```

2. Create another query with a Query Name of *Item* and a SELECT statement of

```
SELECT CUSTID, DESCRIP , ITEMTOT, ORDERDATE, ITEM. ORDID
FROM ORD , PRODUCT , ITEM
WHERE ITEM .ORDID = ORD .ORDID
      AND ITEM.PRODID = PRODUCT.PRODID
ORDER BY CUSTID, DESCRIP, ORDERDATE
```

✓ Parent/Child Relationships

and then press [NextField]. This moves the cursor to the Parent Query 1 entry field. Enter *Customer*. This tells SQL*ReportWriter that the Customer query is a parent (or master in the master/detail relationship) of the Item query. Press [Next Field] twice. This moves the cursor to the Child Columns entry field. You must now specify how the parent query is related to the child, Item. Enter the following so that the bottom portion of your screen contains the information below:

Child Columns	Parent 1 Columns	Parent 2 Columns
Custid	Custid	(Blank)

This tells SQLReportWriter to perform an outer-join with the Item and Customer queries.

3. Create a third query with a Query Name of *Summary* and a SELECT statement of

```
SELECT CUSTID, DESCRIP , SUM (ITEMTOT) TOTAL
FROM ORD , PRODUCT , ITEM
WHERE ITEM . ORDID = ORD.ORD ID
      AND ITEM. PRODID = PRODUCT.PRODID
GROUP BY CUSTID, DESCRIP
ORDER BY CUSTID, DESCRIP
```

Move to the Parent Query 1 entry field and enter *Customer*. Then join the parent and the child on the *custid* column. (The Child Columns and Parent Columns should be identical to the Item query.)

Group Screens

4. Move to Group Screen One and insert a group named `G_Product` below `G_Customer` (press [Insert Record Below] while your cursor is on the `G_Customer` record). Select Item from the List of values for its Query Name. You still need to assign at least one field to the group you just created. You will do so in step 7.
5. Next, enter `Conditional` (or select it from the List of values) for the Page Break setting of the `G_Customer` group. This tells SQL*ReportWriter that if records of the Customer group and its related records do not fit on one page, the entire group (and all its records) should be moved onto the next page.
6. Go to Group Screen Two and enter the following values for the groups
 - `G_Customer`: Spaces Before 2
Record Spacing 2
 - `G_Product`: Spaces Before 2
 - `G_Item`: Spaces Before 2
 - `G_Summary`: Relative position: Below
Lines Before: 1
Spaces Before 0 (enter a zero, otherwise it will be the default Spaces Before 1 space)

Field Screens

7. Move to Field Screen One and assign the `G_Product` group to the `CUSTID2` and `DESCRIP` fields (move to the Group entry field for `CUSTID2`, press [List] and select `G_Product`, then do the same thing for the `DESCRIP` field).
8. Change the default field labels to the following

Field Name	Label
<code>DESCRIP</code>	Product
<code>DESCRIP2</code>	Product Totals

9. Now move to Field Screen Two and change the following Field Widths: `TOTAL` to 10, and `NAME` to 20.
10. Change the Display Format of both the `ITEMTOT` and `TOTAL` fields to `$ZZ,ZZ9.99`. **Note:** The Zs must be capitalized.

11. Move to Field Screen Three and skip the following fields by placing an x in the Skip entry field: CUSTID3, ORDID, CUSTID2, and CUSTID. When you “skip” fields, SQL*ReportWriter will not display them in the report output.

Execute your report.

Group Footing Report (V1.0)

Deptno	Empno	Ename	Sal
10	7782	CLARK	2450
	7839	KING	5000
	7934	MILLER	1300
----- End of Dept. 10 -----			
20	7369	SMITH	800
	7566	JONES	2975
	7788	SCOTT	3000
	7876	ADAMS	1100
	7902	FORD	3000
----- End of Dept. 20 -----			
30	7499	ALLEN	1600
	7521	WARD	1250
	7654	MARTIN	1250
	7698	BLAKE	2850
	7844	TURNER	1500
	7900	JAMES	950

Distinguishing Features

Group Footing reports contain boilerplate text, and/or fields, printed at the end of an instance of a group. In this sample output, the footer of the Employee group, "—End of Dept. fieldvalue---," is printed just before anew value is printed for the Department group (a new department number).

Concepts

To place a group footer in any report, you need to do the following:

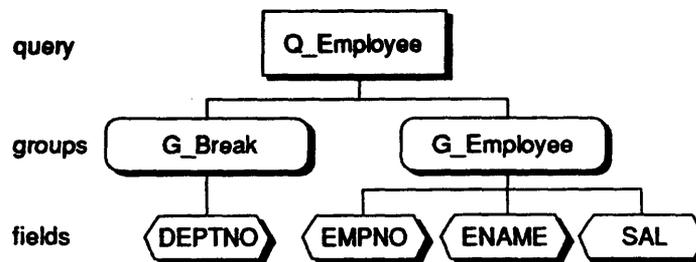
- go to the text screen
- move to the Text entry area (at the bottom of the screen)
- press [Next Record] until you reach a text Object for one of your report's groups, and the text Type is Footer (in matrix reports, go to the Subfooter)
- enter any text you wish (you may also embed fields).

If you embed fields in footers (or headers), make sure that there is only one value for that field in the footer (or header). Otherwise, SQL*ReportWriter will not know which of the several values you want it to print, and it will give an Inconsistent Print Frequency error.

To see a sample Group Footing report, open the SQL*ReportWriter example report named Footing_Report. See the section called "How To Load Sample Reports" on page 4-2 for details.

Organization

One query selects all of the fields in this report, and two groups that belong to that query. The manually created group, G_Break, causes a control break in this report. For information about control breaks, see the Break Report.



Steps to Build this Report

- | | |
|--------------|---|
| Query screen | <ol style="list-style-type: none"> 1. Create a query with a Query Name of Q_Employee and a SELECT statement of

 <pre>SELECT DEPTNO, EMPNO, ENAME , SAL FROM EMP ORDER BY DEPTNO</pre> |
| Group Screen | <ol style="list-style-type: none"> 2. Create a new group (press [Insert Record Above] while your cursor is on G_Employee) and name it G_BREAK. Then press [Next Field] and enter Q_Employee. 3. Move to Group Screen Two and enter a 1 in the Record Spacing entry field for G_Break. |

- Field Screens**
4. Go to the Field Screens and move your cursor until you reach the G_Employee group for the DEPTNO field.
 5. Press [Delete Word] and then enter G_Break. This assigns the DEPTNO field to the group that you created, Break.

✓ **Boilerplate Text**

- Text Screen**
6. Move Your cursor to the Text Screen and press [Next Record] until you reach the screen where the Object is G_Employee and the text Type is Footing. Add the following text

-----End of Dept. &deptno -----

Execute your report.

Page Heading Report (V1.0)

---- EMPLOYEE SUMMARY ----			
D Deptno	Empno	Ename	Sal
-----	-----		
10	7782	CLARK	2450
10	7839	KING	5000
10	7934	MILLER	1300
20	7369	SMITH	800
20	7566	JONES	2975
20	7788	SCOTT	3000
20	7876	ADAMS	1100
20	7902	FORD	3000
30	7499	ALLEN	1600
30	7521	WARD	1250
30	7654	MARTIN	1250
30	7698	BLAKE	2850
30	7844	TURNER	1500
30	7900	JAMES	950

Distinguishing Features

A Page Heading report contains text and/or fields in the report's page header. The heading can span several lines. This example report is a Tabular Report with boilerplate text (any text you enter) in the page header.

For an example of a report that has a database field in its page header, see the "Placing Database Values in Page Headers" report.

Concepts

Specify the Object Type as Page so the heading will print at the top of each page.

To place a page header in any report, you need to do the following:

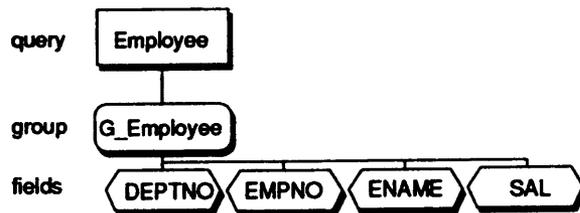
- go to the text screen
- move to the Text entry area (at the bottom of the screen)
- press [Next Record] until you reach the text Object of Page and the text Type is Header
- enter any text you wish (you may also embed fields).

If you embed fields in headers (or footers), make sure that there is only one value for that field in the header (or footer). Otherwise, SQL*ReportWriter will not know which of the several values you want it to print, and it will give an Inconsistent Print Frequency error.

To see a sample Page Heading report, open the SQL*ReportWriter example report named Heading_Report. See the section called "How To Load Sample Reports" on page 4-2 for details.

Organization

One query selects all of the fields in this report. There are no manually created groups.



Steps to Build this Report

Query Screens

1. Create a query with a Query Name of `Employee` and a SELECT statement of

```
SELECT DEPTNO, EMPNO, ENAME , SAL FROM EMP
ORDER BY DEPTNO, EMPNO
```

✓ Page Headers (creating)

Text Settings

2. Move your cursor to the Text Screen and press [Next Record] until you reach the screen where the Object is Page and the Text Type is Header. Change the Justification to center, and then move to the Text entry area and add the following text:

```
---- Employee Summary ----
```

followed by one carriage return (press the Return key do not enter &CR).

Execute your report.

ADVANCED REPORTS

This chapter contains the distinguishing features, concepts, organization, and building steps for each of the following advanced reports:

- Intermixing Fields From Different Groups using the Margin setting
- Creating Groups With No Printable Fields
- Conditional Highlighting
- Relative Positioning Of Unrelated Group using null queries
- Relative Positioning Of Related Groups using null queries
- Suppressing Column Headers When No Detail Records Are Retrieved
- Placing Database Values In Page Headings
- Aggregating Data Within Ranges
- Printing Reports On Preprinted Forms
- Across Reports With Control Breaks
- Wrapped Break Report
- Computations On Summaries using cross-referenced queries
- Computations On Summaries using SQL
- Time Series Calculations
- Advanced Mailing Label Report

Advanced Reports (Continued):

- Invoice Report
- Check Printing Report
- Spelling Out Cash Amounts On Checks
- Matrix Report With Zeros For Null Values
- Matrix Break Report
- Ranking Report
- Changeable Number Of Records per Column Report
- Conditional Printing.

How To Load Advanced Reports

Several advanced reports are shipped with SQL*ReportWriter. This chapter explains those reports in detail by describing the:

- distinguishing features
- concepts
- organization
- steps to build the report.

If you would like to see the advanced reports that are shipped with SQL*ReportWriter, do the following:

1. Type `loadrep advanced. rex username/password` while you are in the SQL*ReportWriter demo directory. Once you have loaded the latest reports using `loadrep advanced.rex`, you need not do it again.

Note: If you already have the advanced reports from SQL*ReportWriter Version 1.1.10 or earlier, you must drop *those* reports before loading the latest reports. To do this, type `drop_adv. com username/password` while in the demo directory. (The command syntax for your operating system may vary. For more information, see your Installation and User's Guide.) Then type `loadrep advanced. rex username/password`.

2. Open the report in the report definer, SQLREP.

Intermixing Fields From Different Groups: using the Margin setting (V1.1)

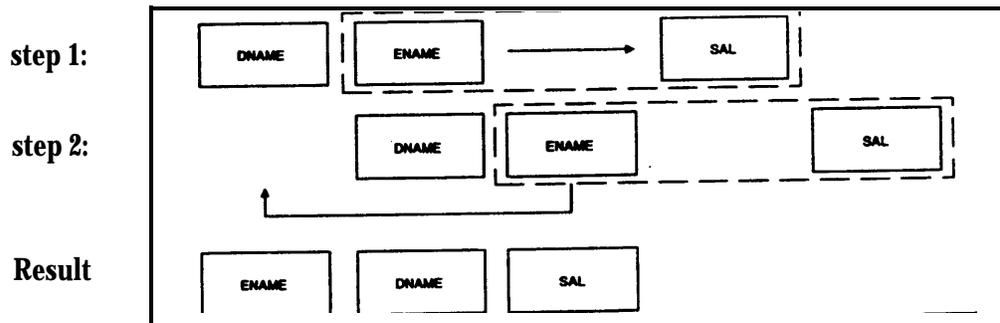
Ename	Dname	Sal
CLARK	ACCOUNTING	2450
KING		5000
MILLER		1300
ADAMS	RESEARCH	1100
FORD		3000
JONES		2975
SCOTT		3000
SMITH		800

Distinguishing Features

Normally, a break field appears to the left of fields that are related to it. In this example, the break field appears between its related fields.

Concepts

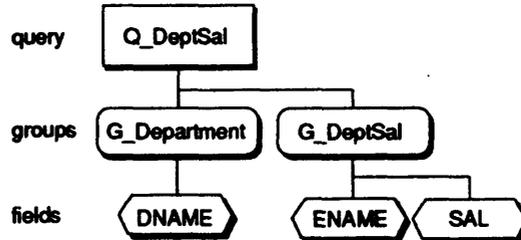
The challenge to building this type of report is to move a field of one group between two fields of another group. To do this, you first add spaces between the two fields of the other group to make room for the field to fit between them (Step 1 below). Next, you add spaces before the first field, and then move the group with the two fields to the left margin using the Margin Relative Position setting on the Text Screen (step 2 below).



To see a sample report intermixing fields, open the SQL*ReportWriter example report named Adv_Intermix_Fields. See the section called “How To Load Advanced Reports” on page 5-2 for details.

Organization

This example report contains one query and two groups: one default group, and one manually-created group. The DNAME field is assigned to the manually-created group—the break group.



Steps to Build this Report

- Query screen 1. Create one query with a query name of Q_Dept Sal and a SELECT statement of

```

SELECT ENAME, SAL, DNAME FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO
ORDER BY DNAME, ENAME
  
```

- Group Screens 2. Create a group above G_DeptSal with a name of G_DeptSal.
3. Enter a 2 for the Record Spacing of G_DeptSal.

- Field Screens 4. On Field Screen One, assign the DNAME field to the G_DeptSal group. Delete its Field Label by using [Delete Word]. On Field Screen Two, enter a 2 in the Lines Before setting for DNAME. On the same screen, enter a 17 for DNAME and a 25 for SAL in the Spaces Before setting.

Text Screen 5. On the Text Screen, move to where the Object is G_DeptSal and the Type is Column Heading. Enter Margin for the Relative Position, and then move to the Text area. Between the ENAME and SAL field labels, type DNAME and a row of hyphens under it.

Move to the next screen (the Object is G_DeptSal and the Type is Body), and enter Margin for the Relative Position

Execute your report.

Creating Groups With No Printable Fields (V1.1)

Empno	Ename	Hiredate	Sal
7788	SCOTT	09-DEC-82	3000
7902	FORD	03-DEC-81	3000
7934	MILLER	23-JAN-82	1300
7369	SMITH	17-DEC-80	800
7876	ADAMS	12-JAN-83	1100
7900	JAMES	03-DEC-81	950
7782	CLARK	09-JUN-81	2450
7566	JONES	02-APR-81	2975
7698	BLAKE	01-MAY-81	2850
7839	KING	17-NOV-81	5000
7499	ALLEN	20-FEB-81	1600
7654	MARTIN	28-SEP-81	1250
7844	TURNER	08-SEP-81	1500
7521	WARD	22-FEB-81	1250

Distinguishing Features

In most reports, you create groups and assign fields to them to move or modify the appearance of fields in your report. Sometimes, however, you may find that you want to create a group that does not display a field in the report output. There are three ways to create a group that does not display a field in the output:

- Select Null from system.dual, set the default Field Width to 1, and delete the default Label. This method prints only one blank space. (All versions of SQL*ReportWriter.)
- Create a field with a Source of #SRW_NO_OP (the pre-packaged user exit that does nothing but return), and change the default Datatype to PRT. This method does not print any space because SQL*ReportWriter thinks that the field you are printing is a printer code, and printer codes have a Field Width of 0. See the SQL*ReportWriter Reference Manual for details on printer codes. (For Version 1.1.10 or later, enter a Source of #RWENOP.)
- Create a new group manually, assign a field to it, but then Skip the field on Field Screen Three. (This applies to Version 1.1.10 or later-for Version 1.1.8 and earlier, a group had to display at least one field. That is no longer true.)

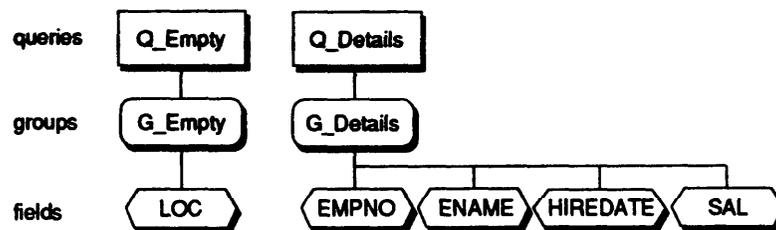
Concepts

In this report, you will create a group that does not display any fields, using the last two methods discussed above. To learn how to create a group that prints the Null field, see the report named, "Relative Positioning of Unrelated Groups: using null queries."

To see a sample report containing groups with no printable fields, open the SQL*ReportWriter example report named Adv_NoPrint_Grps_V11110. See the section called "How To Load Advanced Reports" on page S-2 for details.

Organization

This report is built using two queries and two groups.



Steps to Build this Report

There are two methods that are discussed in the steps below. In Method One, you will use the user exit of Version 1.1.8 and earlier (#slw_no_op). In Method Two you will use the Version 1.1.10 and later functionality (create a group manually, assign a field to it, and then Skip the field).

Method One:

Query Screen

1. Create a query with a name of Q_EMPTY and a SELECT statement of

```
SELECT LOC FROM DEPT
```

2. Create a query with a name of Q_Details and a SELECT statement of

```
SELECT EMPNO , ENAME, HI REDATE, SAL
FROM EMP
ORDER BY JOB, DEPTNO
```

Execute your report so that you can see what it looks like before you change the Datatype of the LOC field to PRT.

Group Screens 3. Enter a 0 in the Spaces Before setting for the G_Details group (otherwise the default two spaces will appear before G_Details is printed).

Field Screens 4. Delete the Source entry field for LOC and enter SRW_NO_OP.
5. Move to Field Screen Two and change the DataType for the LOC field to PRT. (**Note:** You are only able to enter the DataType entry field after you have entered a user exit for the Source of the field.)

Execute your report. Notice that the LOC field did not appear in your report output, and that the EMPNO field is displayed in the exact position that the LOC field had been displayed. (**Note:** You did not need to change the field Width of the LOC field because a field that has a DataType of PRT has a Field Width of 0, regardless of any number specified for the Field Width.)

In Method Two, you will create a group with no printable fields by skipping the only field it owns. (Method Two works only in SQL*ReportWriter Version 1.1.10 or later.)

Method Two:

Query Screen 1. Create a query with a name of Q_EMPTY and a SELECT statement of

```
SELECT LOC FROM DEPT
```

2. Create a query with a name of Q_Details and a SELECT statement of

```
SELECT EMPNO , ENAME, HIREDATE , SAL  
FROM EMP  
ORDER BY JOB, DEPTNO
```

Execute your report so that you can see what it looks like before you skip the LOC field, which will cause the G_Empty group not have any fields to display in the report output.

Group Screens 3. Enter a 0 in the Spaces Before setting for the G_Details group (otherwise the default two spaces will appear before G_Details is printed).

Field Screens 4. Skip the LOC field on Field Screen Three.

Execute your report. You will not see any spaces or blank lines being printed for the G_Master group that does not own any fields, nor will you get an error stating that you must assign at least one field to the empty group you created.

Conditional Highlighting (V1.1)

Empno	Ename	Sal
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7566	JONES	2975
7654	MARTIN	1250
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7839	KING	5000
7844	TURNER	1500
7876	ADAMS	1100
7900	JAMES	950
7902	FORD	3000
7934	MILLER	1300

Distinguishing Features

Conditional Highlighting reports highlight database values and/or text when certain criteria are met. You use this type of report when you want database values to determine what parts of the report should be highlighted. In this example, SQL*ReportWriter bolds the names and salaries of all employees whose salaries are greater than 2000.

Concepts

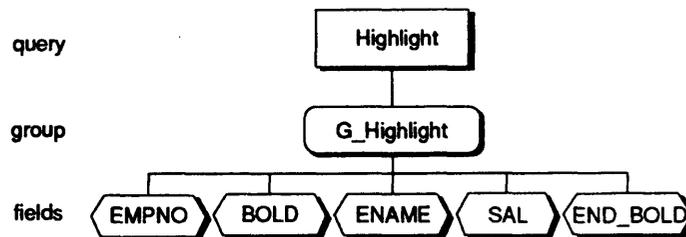
To produce this report, you add a format code (an escape sequence for the printer to perform) to the printer definition for the printer you will use to print the report. (The SQL*ReportWriter default printer definition file is PRINTDEF.DAT.) Next, you manually create two new fields and place them around the fields you want to conditionally highlight. The Sources of these fields are SQL*ReportWriter pre-packaged user exits (RWEIF for Version 1.1.10 or later, SRW_IF for Version 1.1.8 or earlier) that evaluate the database value. If the value meets the condition in the user exit, the user exit field is given the value of the printer code. Otherwise, it is given the value of NULL, and no highlighting occurs. The second user exit field turns the highlighting off by always having a "normal" printer code as its value.

In the example, the user exit field compares employees' salaries. If an employee's salary is equal to or greater than 2000, then the bold format code is the assigned value for the field before the employee name and salary fields (the field is not visible in the output), and the normal format code turns the highlighting off after the instance.

The example report named Adv_Cond_Highlight displays on terminal screens and does not show conditionally highlight fields. See the section called "How To Load Advanced Reports" on page 5-2 for details on loading this report.

Organization

One query selects all of the fields for this report. There are no manually created groups.



Steps to Build this Report

✓ Modifying PRINTDEF.DAT

1. Invoke a text editor and then open the PRINTDEF.DAT file. (see your Installation and User's Guide to learn the location of this file.)
2. Move to the DEC generic printer driver, shown below:

```
DC IDEC IDEC printers:\
:ff="L:cr="M:nl="J:is= ^L:so=\E[lttt:se=\E[Om:li#66:co#80
:ap:us=\E[4m:ue=\E[Om:
```

3. Move to the end of the DEC driver and add the following string to the second line:

```
1=\E[1m:2=\E[Om:
```

The string makes "1" a printer code for holding text, and "2" a printer code for normal text.

4. Save the PRINTDEF.DAT file as DECBOLD.DAT, and exit the text editor.

5. At the operating system command prompt, enter the following

```
printdef dec decbold decbold.dat
```

This compiles the new decbold.dat printer driver, and saves the compiled file as decbold. (See the SQL*ReportWriter Reference Manual for the PRINTDEF syntax, if necessary.)

- Query Screen
6. Enter `sqlrep` and in a new report create a query named Highlight with the following SELECT statement

```
SELECT EMPNO, ENAME , SAL FROM EMP
```

4 User Exit (*rweif*)

- Field Screens
7. Between EMPNO and ENAME, create a new field named BOLD. Enter the following for its Source

```
#rweif sal >= "2000" bold "1""
```

(Note: For Version 1.1.8 or earlier, substitute SRW_IF for RWEIF.)

This user exit says the following: "If an employee's salary is greater than or equal to 2(X)(), place a 1 in BOLD. Otherwise, leave BOLD empty." The 1 will turn on bold highlighting.

Assign the new field to the G_Highlight group.

8. Change the DataType of BOLD to PRT (printer code), and enter a Field Width of 1. (Regardless of the Field Width you enter, a field with a DataType of PRT will have a width of zero and a height of one.)
9. After SAL create another new field named END_BOLD. Enter the following for its Source

```
#rweif sal >= "2000" end bold "2""
```

This user exit is identical to the first one, except that if the salary is greater than or equal to 2000, a 2 is placed in END_BOLD. The 2 will turn off bold highlighting.

Assign the new field to the G_Highlight group.

10. Change the DataType to PRT, and enter a Field Width of 1.
11. Enter a 0 for the Spaces Before setting for both the ENAME and END_BOLD fields.

Parameter Screens 12. Enter the following Default Values:

Parameter Name	Default Value
DESTYPE	Printer
DESNAME	the name of your DEC punter
DESFORMAT	DECBOLD

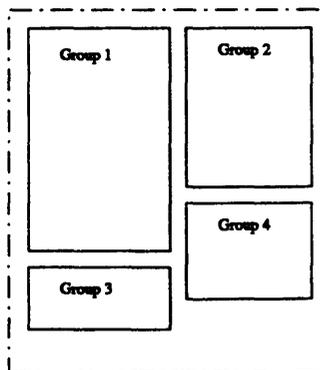
Execute your report. Your printer output highlights in bold records with salaries greater than 2000. **Note:** you will not see the conditional highlighting if you send the report to the screen.

Relative Positioning Of Unrelated Groups: using null queries (V1.0)

Name	Job
EVERY MOUNTAIN	ANALYST
JOCKSPORTS	CLERK
JUST TENNIS	MANAGER
K + T SPORTS	PRESIDENT
NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER	SALESMAN
SHAPE UP	
TKB SPORT SHOP	Dname
VOLLYRITE	
WOMENS SPORTS	ACCOUNTING
	OPERATIONS
Loc	RESEARCH
	SALES
BOSTON	
CHICAGO	
DALLAS	
NEW YORK	

Distinguishing Features

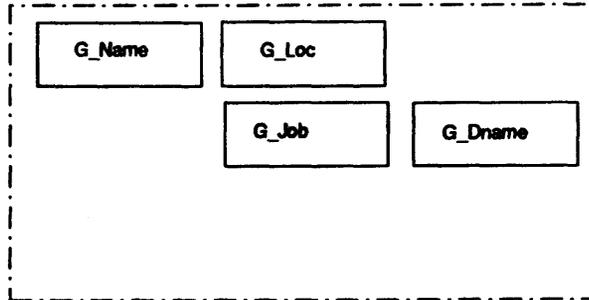
In SQL*ReportWriter, groups maybe positioned to the Right, or Below other groups. This report shows how you can simulate a Relative Position of "Left." This technique enables you to print the Job group to the right of the Name group, the Dname group below the Job group, and the Loc group below the Name group. This report is formatted correctly, regardless of the number of rows fetched for each group.



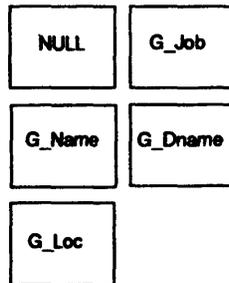
Concepts

The challenge of this example report is to print G_Loc below G_Name. To accomplish this, you must move the query associated with G_Loc down in the query hierarchy by creating an extra query (in this example, the Null query).

The default Relative Position of groups for Down and Down/Across reports is *Right*. Therefore, if this same report was executed without the Relative Positions being modified, this report output would be:



When the Null query is created, it has the following impact on the rest of the group

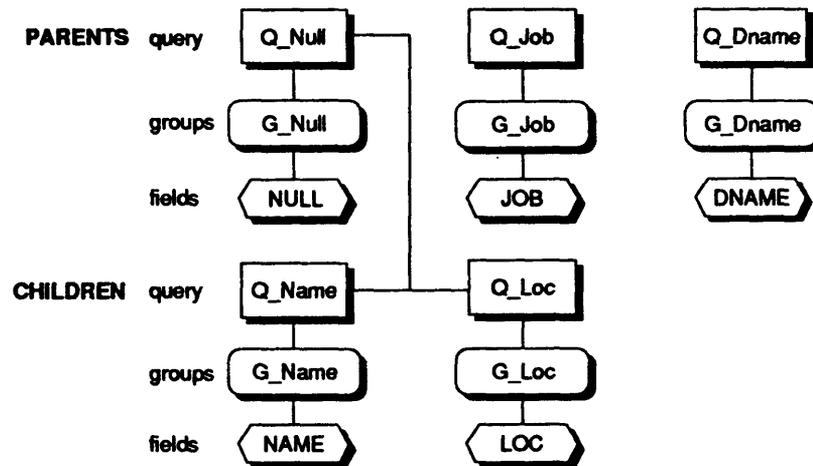


- G_Name begins printing on the same line as its parent, Null. Even though the NULL field does not seem to print (only one space is “displayed”), G_Job appears directly to the right of Null. For this reason, if you do not put a 1 in the Lines Before setting for the G_Job group, G_Job will appear one line above G_Name.
- G_Loc prints one line below and at the left margin of its parent, Null.

To see a sample report showing the relative positioning of unrelated groups using null queries, open the SQL*ReportWriter example report Mined Adv_Pos_UNrelated_Grps See the section called “How To Load Advanced Reports” on page 5-2 for details.

Organization

This example report is built using five queries Each query selects one field; one of the five queries selects Null from SYSTEM.dual. Note that each query may select several fields, but only one is used in this example for simplicity. The Q_Null query is assigned to be the parent of two children, Q_Name and Q_Loc, with no columns related.



The groups appear in the following order on the group screens (top to bottom): G_Null, G_Name, G_Loc, G_Job, G_Dname.

Steps to Build this Report

- Query Screen
1. Create the following five queries Null, Name, Lot, Job, **and** Dname.
 - Null: `SELECT NULL FROM SYSTEM. DUAL`
 - Name: `SELECT DISTINCT NAME FROM CUSTOMER`
Join the Name query to its parent, null, without specifying any columns on which to pin.
 - Loc: `SELECT DISTINCT LOC FROM DEPT`
Join Loc to its parent, Null, without specifying any columns on which to pin.

- Job: SELECT DISTINCT JOB FROM EMP
- Dname: SELECT DISTINCT DNAME FROM DEPT

Group Screens

2. Enter the following Relative Position settings on Group Screen Two:

✓ Relative Position of groups

Group Name	Relative Position	(Result)
G_NULL:	(Blank)	(places it in the top-left corner by default)
G_Name	(Blank)	(places it to the right of G_Null)
G_Loc:	Below	(places it below G_Name)
G_Job	(Blank)	(places it to the right of G_Name by default)
G_Dname:	Below	(places it below G_Job)

Field Screens

3. On Field Screen One, delete the Field Label for the NULL field to prevent it from appearing. If you did not delete the Field Label of Null, the field would print as

```
NULL
----
```

with ten blank spaces printed directly below the four hyphens.

You do not use the Skip setting (on Field Screen Three) to prevent the field from appearing because each group must contain one displayed field in SQL*ReportWriter Version 1.1.8 or earlier.

4. Change the default Field Width of the NULL field to 1.

✓ DataType of PRT

If you do not like the single space results (because Field Widths of NUM, DATE, and CHAR fields must be at least 1), you can assign the Null field to have a DataType of PRT, or Skip the Null field (using SQL*ReportWriter Version 1.1). See the example report in this book entitled "Creating Groups With No Printable Fields" for details.

Execute your report. You may want to adjust the Lines Before, Spaces Before, and Record Spacing settings to improve the appearance of your report.

Relative Positioning Of Related Groups: using null queries (V1.0)

Ordid	Orderdate	Shipdate	Custid	Name	Rep id
610	07-JAN-87	08-JAN-87	101	TKB SPORT SHOP	7521
Prodid	Ordid				
100860	610				
100870	610				
100890	610				

Ordid	Orderdate	Shipdate	Custid	Name	Rep id
611	11-JAN-87	11-JAN-87	102	VOLLYRITE	7654
Prodid	Ordid				
100861	611				

Distinguishing Features

This report is similar to the one shown on the previous page, except that the groups are related. Therefore, the groups in the lighter rounded rectangles are children of the group in the darker rounded rectangle.

Concepts

In this report, you want to position data as shown below. The positioning is similar to the Relative Position of Unrelated Groups; however, this report has a twist-groups 1,2, and 3 are related.

The challenge in this example report is to place G_Items directly below G_Orders, *and* to have all three groups joined so that for each order the correct items and customer name is displayed. To accomplish this, you must

- move G_Items down in the group hierarchy by changing its parent from G_Order to G_Null so that it can be placed below G_Orders, and
- join G_Items and G_Orders via their parent G_Null, and join G_Name to G_Orders.

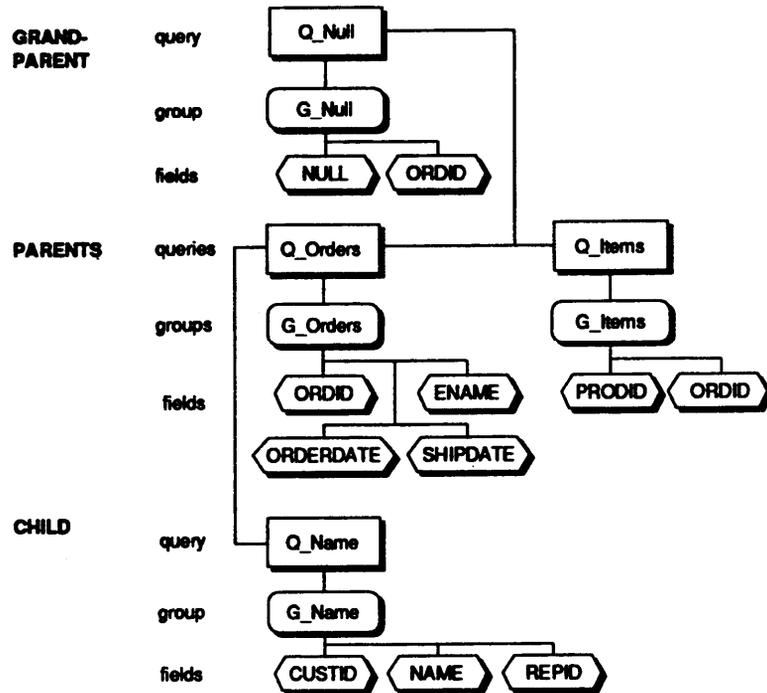
In this example, therefore, you will

- assign G_Orders and G_Items to share a “null” parent (G_Null), and
- specify columns to pin G_Orders and G_Items to G_Null, and
- assign G_Name to be the child of G_Orders and specify a column to pin them.

To see a sample report showing the relative positioning of related groups using null queries, open the SQL*ReportWriter example report named Adv_Pos_Related_Groups. See the section called “How To Load Advanced Reports” on page 5-2 for details.

Organization

In the following example, you will enter four queries one query for each section (or group), and one "Null" query.



Steps to Build this Report

- Query Screen
1. Create the following four queries: `Q_Null`, `Q_Orders`, `Q_Name`, and `Q_Items`.
 - `Q_Null`: `SELECT NULL, ORDID FROM ORD ORDER BY ORDID DESC`
 - `Q_Orders`: `SELECT ORDID, ORDERDATE, SHIPDATE, CUSTID FROM ORD`

Join `Q_Orders` to its parent, `Q_Null`, and specify the `ORDID` column for the pin.

•Q_Name: SELECT CUSPID, NAME, REPID FROM CUSTOMER

Join Q_Nametoitsparent, Q_Orders, and specify the CUSTID column for the pin.

•Q_Items: SELECT PRODID, ORDID FROM ITEM

Join Q_Items to its parent Q_NULL, and specify the ORDID column for the pin.

Group Screens 2. Enter the following Relative Position setting on Group Screen Two:

✓ Relative Position of groups

Group Name	Relative Position	(Result)
G_Null:	(Blank)	(places it at the top-left of the report)
G_Orders:	Below	(places it below G_Null)
G_Name	(Blank)	(places it to the right of G_Orders)
G_Items:	Below	(places it below G_Orders because G_Items and G_Orders are siblings)

Field Screens On Field Screen One, delete the Field Label for NULL (press [Delete Word], do not space over the Label).

On Field Screen Three, "Skip" the ORDID and CUSTID2 fields.

Execute the report. You may want to adjust the Lines Before, Spaces Before, and Record Spacing settings to improve the appearance of the report.

Suppressing Column Headings When No Detail Records Are Retrieved (V1.0)

Deptno 20	Ename	Job
Dname RESEARCH	-----	-----
Loc DALLAS		
	SMITH	CLERK
	JONES	MANAGER
	SCOTT	ANALYST
	ADAMS	CLERK
	FORD	ANALYST
Deptno 30	Ename	Job
Dname SALES	-----	-----
Loc CHICAGO		
	ALLEN	SALESMAN
	WARD	SALESMAN
	MARTIN	SALESMAN
	BLAKE	MANAGER
	TURNER	SALESMAN
	JAMES	CLERK
Deptno 40		
Dname OPERATIONS		
Loc BOSTON		

Distinguishing Features

In Master/Detail reports, the column headers appear for the detail group, even if no detail records are retrieved. This report shows how to suppress printing of these column headers when there are no associated details. For example, the column headers (Empno and Job) of the detail group are not displayed because there are no employees in department 40.

The techniques described in this report can also be used to suppress detail group summaries when no records are retrieved for the detail group.

Concepts

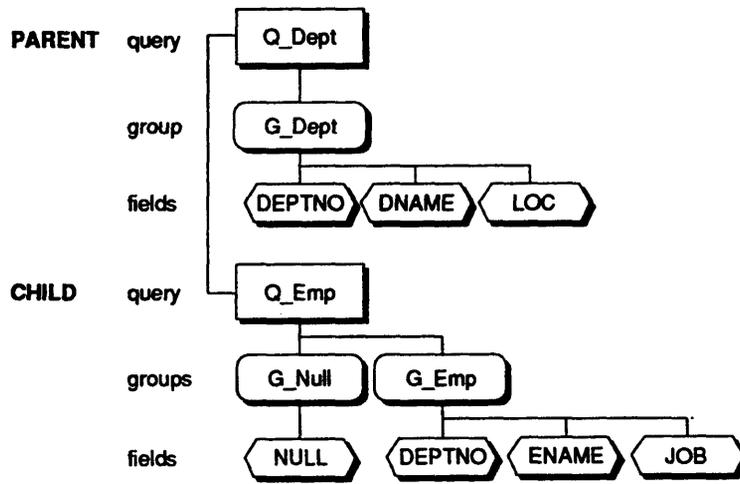
For Master/Detail reports, it is a rule that whenever a master record prints, a detail label or field must also print, even if there aren't any fetched detail records. This is because some users have expressed that they want the labels to appear even if no records are retrieved. However, because some users do not want the label to appear when there aren't any fetched records, this report can be built.

Although it appears as if nothing is displayed where the detail label generally appears, a field's contents is displayed: the NULL field's contents (one space). A group was inserted above the detail's default group, and the NULL field was assigned to the new group (and the label for the NULL field was deleted). When the NULL field's contents is printed, the rule is followed, and the null group's child is not required to be displayed.

To see a sample report that suppresses column headings when no detail records are retrieved, open the SQL*ReportWriter example report named Adv_Suppress_Col_Hdrs. See the section called "How To Load Advanced Reports" on page 5-2 for details.

Organization

This example report is built with two queries, Q_Dept and Q_Emp, that share a master/detail relationship. A third group, G_Null, is created manually and is associated with the Q_Emp query.



Steps to Build this Report

Query Screen 1. Enter the following two queries Q_Dept , and Q_Emp .

- Q_Dept: SELECT * FROM DEPT ORDER BY DEPTNO
- Q_Emp: SELECT DEPTNO, ENAME, JOB, NULL FROM EMP ORDER BY ENAME

Join Q_Emp to its parent, Q_Dept , via the DEPTNO columns.

- | | |
|-----------------------------------|--|
| Group Screens | <ol style="list-style-type: none"> 2. Insert a new group above G_Emp, name it G_NULL and associate it with Q_Emp. 3. On Group Screen Two, enter a 1 for the Record Spacing entry field of G_Dept, and a 1 in Fields Across entry field for G_Dept. 4. Set the Label Position for G_Dept to Left. |
| ✓ Field labels (modifying) | |
| Field Screens | <ol style="list-style-type: none"> 5. Assign the NULL field to the G_Null group. You do this to meet the (SQL*ReportWriter Version 1.1.8 and earlier) requirement that all groups must have and print at least one field. In this report, the NULL field prints ten blank spaces.

If you do not like the spaces that result from the NULL field being printed and you are using SQL*ReportWriter Version 1.1.8 or later, see the example report in this book entitled, "Creating Groups With No Printable Fields: to learn how to remove the spaces. . 6. Delete the label of the NULL field so that it doesn't appear in the report output. 7. Suppress DEPTN02 from printing by Skipping it on Field Screen Three. |
| Text Screen | <ol style="list-style-type: none"> 8. Enter a Frequency of G_NULL for the G_Emp Column Header text object. You do this to tie the appearance of the Column Header to the appearance of the G_Null group. If there are no employees, as is the case for department 40, the NULL field is not printed, so the G_Emp Column Header text object does not print either. <p>Execute your report.</p> |

Placing Database Values In Page Headings (V1.1)

Departments found on this page: 10 through 20				Departments found on this page: 20 through 30		Departments found on this page: 30 through 30	
Deptno	Ename	Job	Sal	Sal	Sal	Sal	Sal
10	CLARK	MANAGER	2450	3000	800	1500	1250
	KING	PRESIDENT	5000	1600	2850		
	MILLER	CLERK	1300	950			
20	ADAMS	CLERK	1100	1250			
	FORD	ANALYST	3000				
	JONES	MANAGER	2975				

Distinguishing Features

One or more database values can be located in the report's header or page header. In this example report, the first and last department number found on each page is displayed in the page header.

Concepts

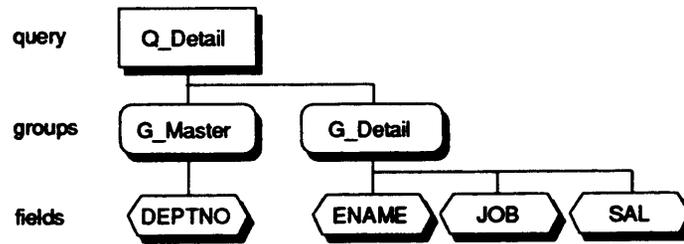
To build this report, you will reference two computed fields in a report Page Header. You can explicitly reference one or more fields or summaries in a page and/or report Header and/or Footer via two new SQL*ReportWriter Version 1.1 functions: First and Last.

First is the first database value selected for a group, page, or the report. Likewise, Last is the last database value selected. In this report, you will assign one computed field to have a First Function, and a second to have a Last Function. Then all you have to do is enter the computed field references and some boilerplate text in the page header of the report. (In this report, the page length was shortened to illustrate how the page numbers change dynamically, based on the first and last database value fetched on each page.)

To see a sample report that places database values in page headings, open the SQL*ReportWriter example report named Adv_DB_Vals_in_Pg_Hdrs. See the section called “How To Load Advanced Reports” on page 5-2 for details.

Organization

This example report is built with one query and two groups. G_Master, the manually created group, is the break group.



Steps to Build this Report

Query Screen 1. Enter the following SELECT statement in a query called Q_Detail:

```
SELECT DEPTNO , ENAME, JOB, SAL
FROM EMP
ORDER BY DEPTNO, ENAME , SAL
```

Group Screens 2. on Group Screen one, insert a new group named G_Master above the default group and assign it to the Q_Detail query.

3. On the next screen, set the Record Spacing for G_Master to 1.

Field Screens 4. Assign the DEPTNO field to the G_Master group.

5. Change the Field Width of the DEPTNO field to 2.

- Summary Screens
- ✓ First Function
 - ✓ Last Function
6. Create two summary fields: `FIRSTDEPTREC`, and `LASTDEPTREC`.
- Assign `FIRSTDEPTREC` to summarize the `DEPTNO` field, give it a Function of `First`, a Print Group of `G_Detail`, and a Reset Group Of Page.
 - Assign `LASTDEPTREC` to summarize the `DEPTNO` field, give it a Function of `Last`, a Print Group of `G_Detail`, and a Reset Group Of Page.
- Text Screen
7. Go to the Page Header text object, enter Center for the Justification setting, and then enter Departments found on this page: `&FirstDeptRec` through `&LastDeptRec` (the text for the page header and the field references).
- Execute your report.
8. Change the report length from 66 to 10 and execute the report again. Page through the report to see how the values in the page header change dynamically. Change the report length to a smaller or larger number to see how this affects the values in the page headers.

Aggregating Data Within Ranges (V1.0)

Salary Range	Ename	Deptno
	SMITH	20
	JAMES	30
0 - 999		
: 1000 - 1999	ADAMS	20
	WARD	30
	MARTIN	30
	MILLER	10
	TURNER	30
	ALLEN	30
2000 - 2999	CLARK	10
	BLAKE	30
	JONES	20

Distinguishing Features

Values from the database are retrieved and formatted based on any aggregate range that you define. In this example report, the aggregate range is in increments of one thousand, starting at zero.

Concepts

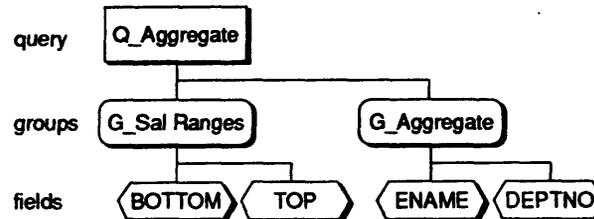
It is fairly simple to create reports that aggregate data over pre-specified ranges. You can even use parameters to specify the range in which the data should be aggregated. In this report, you use a SQL statement with two "functions."

The functions are $(\text{FLOOR}((\text{SAL} + 1)/1000) * 1000)$, which calculates the lowest salary, and $(\text{CEIL}((\text{SAL} + 1)/1000) * 1000 - 1)$, which calculates the highest salary. The fields that receive the values of the functions are placed into a break group to produce the above control break format.

To see a sample report that aggregates data within ranges, open the SQL*ReportWriter example report named Adv_Aggregating_Data. See the section called "How To Load Advanced Reports" on page 5-2 for details.

Organization

This report has the organization of a simple break report. The fields that comprise the salary ranges are placed into the break group.



Steps to Build this Report

Query Screen 1. Create a query called `Q_Aggregate` that has a SELECT statement of

```
SELECT FLOOR ( (SAL + 1) /1000) * 1000 BOTTOM,
CEIL ((SAL+ 1)/1000)*1000- 1TOP,
ENAME, DEPTNO
FROM EMP
ORDER BY 1, 2, SAL
```

Group Screens 2. Create a second group called `G_SalRanges` above the default group.

Field Screens 3. Go to Field Screen One and assign the `BOTTOM` and `TOP` fields to the group `G_SalRanges`.

4. On Field Screen Two, change the Field Width of `BOTTOM` and `TOP` to

5. On Field Screen Three, set the Align setting for `BOTTOM` to Left.

Text Screen 6. Edit the Column Heading of `G_SalRanges` so that it looks like:

```
Salary Range
-----
```

and the text Body of `G_SalRanges` so that it looks like:

```
&BOTTOM - &TOP
```

Execute your report. You may want to assign a Record Spacing of 1 for the `G_SalRanges` group.

Printing Reports On Pre-printed Forms (V1.1)

Summit Sporting Goods, Inc.
123 Main Street
Anytown, CA 12345

Sales Order No. 068942

Salesrep ALLEN
Customer EVERY MOUNTAIN
574 SURRY RD.
CUPERTINO CA 93301

Date	Code	Product Description	Quantity	Price	Amount
18-JUL-86	100871	ACE TENNIS BALLS-6 P	1	\$5.60	\$5.60
25-JUL-86	100871	ACE TENNIS BALLS-6 P	2	\$5.60	\$11.20
	101860	SP TENNIS RACKET	1	\$24.00	\$24.00
15-JAN-87	100860	ACE TENNIS RACKET I	100	\$30.00	\$3,000.00
	100861	ACE TENNIS RACKET II	20	\$40.50	\$810.00
	100871	ACE TENNIS BALLS-6 P	100	\$5.50	\$550.00
	101863	SP JUNIOR RACKET	150	\$10.00	\$1,500.00
22-FEB-87	100871	ACE TENNIS BALLS-6 P	50	\$5.60	\$280.00
	102130	RH: "GUIDE TO TENNIS	100	\$3.40	\$340.00
	200376	SB ENERGY BAR-6 PACK	100	\$2.40	\$240.00
	200380	SE VITA SNACK-6 PACK	100	\$4.00	\$400.00

Item totals \$7,160.80

Distinguishing Features

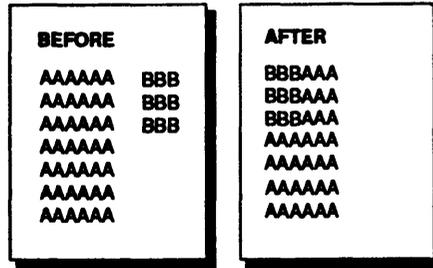
This example report illustrates report formatting techniques for printing reports on pre-printed forms. The report outlines the design methodology for printing reports on a company sales order form, in this case a mock up of a form for a fictitious company named Summit Sporting Goods.

Concepts

Reports that print on pre-printed forms must be designed so that the report's data is printed in exact positions on the form. In this example, the form length is 24 lines per page; the customer name and address must begin on line 2, column 15; the customer name must word-wrap if it is longer than 15 characters; and the page item totals must print on line 22. This report is designed so the no more than 12 item detail lines will print on each invoice page, the rest spilling over to subsequent pages as needed.

✓ Relative Position of Margin

To build this report, you will use the Relative Position of *Margin* on the Text Screen for several text objects. *Margin* causes a text object to move to the left margin, regardless of any other text object that maybe in its way. See the illustration below.



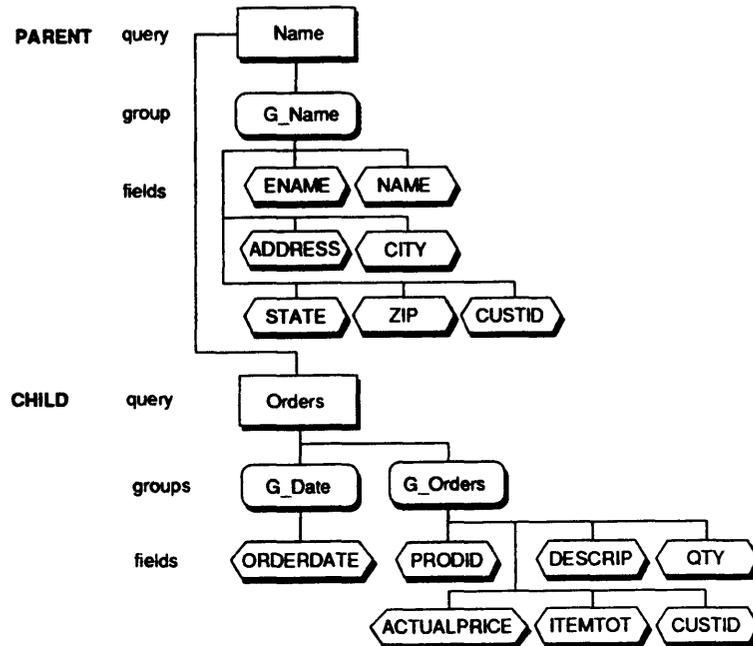
The Margin setting on a text object also changes the reference point from which the position of that object is computed. In the figure above, prior to Setting Margin, the Relative Position of Group B was computed from the right-most edge of Group A. The Margin setting causes SQL*ReportWriter to place the Group B relative to the top corner of the Group A at the left margin. Thus, once a text object is set to Margin, it can be placed anywhere in the report with the Lines Before and Spaces Before settings on the Group and Text Screens.

You will also assign a Page Break setting of Always for the query that selects both the sales rep. and customer address. This will cause each customers orders and expenses to appear once perform.

To see a sample report that prints on pre-printed forms, open the SQL*ReportWriter example report named Adv_PrePrinted_Forms. See the section called "How To Load Advanced Reports" on page 5-2 for details.

Organization

This report is built with two queries, Name and Orders. The queries share a master/detail (or parent/child) relationship: the query called Name is the parent of the Orders query. There is one manually-created group: G_Date.



Steps to Build this Report

Query Screen 1. Create two queries Name, and Orders with the following SELECT statements.

- Name :

```
SELECT ENAME, NAME, ADDRESS,
CITY, STATE, ZIP, CUSTID
FROM EMP , CUSTOMER
WHERE EMPNO = REP ID
ORDER BY ENAME, NAME
```

- Orders:

```
SELECT ORDERDATE, I.PRODID, DESCRIP, QTY,
ACTUALPRICE, ITEMTOT, CUSTID
FROM ORD O, ITEM I, PRODUCT P
WHERE O.ORDID = I.ORDID
AND I. PRODID = P.PRODID
ORDER BY ORDERDATE
```

Join the Orders query to its parent, Name, using the CUSTID columns.

Group Screens

Set the PageBreak for G_Name to Always, so that each customer prints on a new page.

Insert a group above G_Orders, and name it G_Date. Assign the new group to the orders query. In conjunction with Step 5 below, this will cause the detail section to “break on Orderdate.

Move to the Lines Before entry field for G_Date and enter a 1.

Field Screens

Assign the ORDERDATE column to the G_Date group (instead of G_Orders). This causes only one order date to be printed for all of its related items.

Delete all of the field labels. Remember, use [Delete Line]; do not space over the labels. You don’t want the field labels to be displayed because the pre-printed form provides you with its own labels.

On Field Screen Two, change the following fields to the Field Width specified:

Field Name	Field Width
NAME	15
ADDRESS	15
CITY	10
DESCRIP	20
QTY	4
ACTUALPRICE	6

Enter \$Z9.99 for the Display Format of the ACTUALPRICE field, and \$ZZ, ZZ9.99 for the Display Format of the ITEMTOT field.

9. Move to the Relative Position entry field and enter Below for the NAME, ADDRESS, and CITY fields.
10. On Field *Three, enter Variable in the Align entry field for both the NAME and CITY fields, and then Skip the CUSTID and CUSTID2 fields (place an x in the Skip entry field). A Variable alignment causes all padded spaces of a field to be deleted. In the reverse case, if the text object width is less than the field's contents, the contents will word-wrap within the text object width.

Summary Screens 11. Create a summary named `SUMITEMTOT` with the settings described below

summary Name	Field	Function	Width	Display Format	Print Group	Reset Group
SUMITEMTOT	ITEMTOT	SUM	10	SZZ,ZZ9.99	G_Name	Page

Report Screen 12. Change the Page Height setting to 24 lines per page, and the Top Margin setting to 0.

Text Screen 13. In the text object with Page as the Object and Footer as the Type, enter a 56 in the Spaces Before entry field, and then enter:

```
&SUMITEMTOT
```

in the Text entry area.

14. Move to the G_Name Body text object and enter a 2 for the Lines Before setting, and a 15 for the Spaces Before setting, and then modify the Text entry area looks like the following

```
&ENAME &CR
&CR
&NAME2 &CR
&ADDRESS &CR
&CITY, &STATE & ZIP &CR
```

15. Move to the G_Date Body text object and enter a Relative Position of Margin, and then enters 7 for the Lines Before Setting.

16. Finally, move to the G_Orders Body text object and enter a Relative Position of margin, and a 10 for the Spaces Before setting.

Execute your report.

Across Reports With Control Breaks (V1.0)

Deptno	10					
Dname	ACCOUNTING					
Loc	NEW YORK					
Ename	CLARK	KING	MILLER			
	20					
	RESEARCH					
	DALLAS					
	ADAMs	FORD	JONES	SCOTT	SMITH	
	Text					
	Text					
	T	e	x	t		
	T	e	x	t	MARTIN	TURNER

Distinguishing Feature

An Across report prints database values of a column across the page instead of down (see the Ename values in the report above). Across reports are different from Down reports in that the x and y coordinates are reversed.

In Across reports with breaks, the master (or break) group prints "top to bottom: "i.e., as it would in other master/detail reports. However, the values in the detail group prints across the page, instead of top to bottom. Thus, while DEPTNO, DNAME, and LOC print top to bottom, ENAME values print left to right in the example shown above.

Concepts

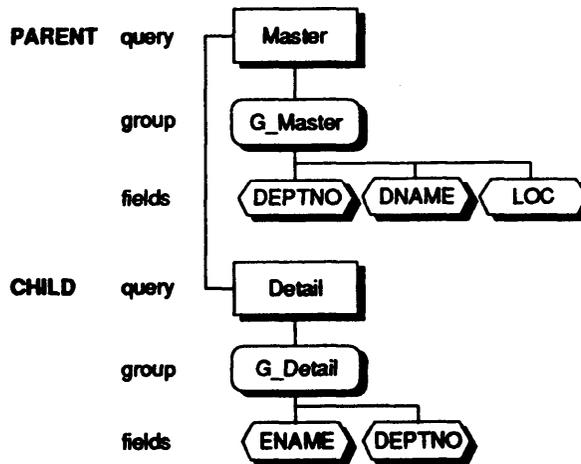
To print field values across the page in your report, you can use the Across Print Direction (on Group Screen One); to print field values down the page in your report, you can use the Down Print Direction. However, you cannot mix Across and Down printing in one report. That is, if you select a Print Direction of Across for one of the groups in your report, you must select either the Across or Across/Down Print Directions for all other groups in that report. Likewise, if you select a Print Direction of Down for one of the groups in your report, you must select either Down or Down/Across Print Directions for the other groups.

This report was built by simulating a Print Direction of Down for the master (i.e., control-break) group. We set the Print Direction of the master group to Across/Down, and then make the group so wide that only one record can fit in the width of the page. This forces the master group to print in a down direction. The detail group prints Across/Down in the normal, expected fashion.

To see a sample across report with control breaks, open the SQL*ReportWriter example report named Adv_Across_Breaks. See the section called "How To Load Advanced Reports" on page 5-2 for details.

Organization

There are two queries in this report that select all of the fields displayed. The queries share a master/detail relationship. There are no manually-created groups.



Steps to Build this Report

Query Screen 1. Create the following two queries:

- Master: `SELECT * FROM DEPT`
- Detail: `SELECT ENAME, DEPTNO FROM EMP
ORDER BY ENAME`

Join Detail to its parent query, `Master`, with the `DEPTNO` column.

- Group Screens
2. Assign a Print Direction of Across/Down for both groups (on Group Screen One).
 3. Enter a 0 for the Lines Before setting of the G_Detail group.
 4. Set the Label Positions for both groups to Left.

- Field Screens
5. Set the Field Width of DEPTNO to 60. This will prevent SQL*ReportWriter from placing more than one value of DEPTNO on one line. On the same screen, enter a 2 for the Lines Before setting for DEPTNO.
 6. On Field Screen Three, enter an Align setting of Left for the DEPTNO field.
 7. Suppress printing of DEPTNO2 (since it already appears in G_Master).

Execute your report.

Wrapped Break Report (V1.1)

Name	Total	Pet.	Total Customers
EVERY MOUNTAIN	5860.00		5.66%
	1260.00		1.22%
	5.60		0.01%
	35.20		0.03%
Sum	7160.80		06.91%
K + T SPORTS	46370.00		44.76%
Sum	46370.00		44.76%
NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER	6400.00		6.18%
Sum	6400.00		06.18%

Distinguishing Features

This report is similar to the Break report because it contains a control break. The difference is that the break field is “wrapped” on word boundaries if it is too long to fit on one line.

If the contents of the break field is longer than the specified Field Width (e.g., North Woods Health...), the field’s contents will not be truncated; instead, the contents wrap within the specified Field Width. If the contents of the break field is shorter than the Field Width (e.g., K + T Sports), the contents do not wrap.

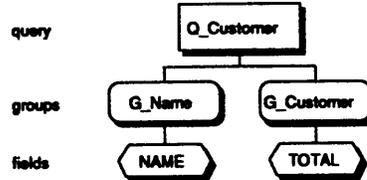
Concepts

This example report is a simple control break report with a summary, and a wrapped field that is located in the break group. The Align setting of Wrap is one of the new features of SQL*ReportWriter Version 1.1. Wrap causes a field’s contents to be formatted onto multiple lines if there are more characters than the Field Width specified.

To see a sample Wrapped Break report, open the SQL*ReportWriter example report named Adv_Wrapped_Break. See the section called “How To Load Advanced Reports” on page 5-2 for details.

Organization

This report has the organization of a simple break report: one query and two groups (one of which is created manually). In this example report, only one field belongs to each group; however, as usual, any number of fields may be used.



Steps to Build this Report

Query Screen 1. Create a query with a name of `Q_Customer` and a SELECT statement of

```
SELECT TOTAL, NAME FROM ORD O, CUSTOMER C
WHERE O. CUSTID = C. CUSTID
ORDER BY NAME
```

Group Screens 2. Create a group above `G_Customer`, name it `G_Name`, and assign it to the `Q_Customer` query. Then move to Group Screen Two, and enter a Record Spacing of 2 for `G_Name`.

Field Screens 3. Assign the `NAME` field to the `G_Name` group.
4. For the `TOTAL` field, enter a Display Format of `ZZZZZ9.99`, and a 4 in the Spaces Before entry field.
5. Change the Field Width of `NAME` to 15, and enter an Align setting of Variable.
6. Create a computed field, `PCT_TOTAL_CUSTOMERS`, that has a Source of `TOTAL`, a Group of `G_Customer`, a Display Format of `ZZZZZ9.99`, a function of `%Total`, and a Reset Group of Report.

Summary screens 7. Create the following two summaries:

✓ Display Formats (altering)

- `CUSTOMER_TOTAL`, that sums the `TOTAL` field, has a Display Format of `ZZZZZ9.99`, and a Print and Reset Group of `G_Name`
- `CUSTOMER_PCT_TOTAL` that sums the `PCT_TOTAL_CUSTOMERS` field, has a Display Format of `ZZZZ9.99%`, and a Print and Reset Group Of `G_Name`.

Execute your report.

Computations On Summaries: using cross-referenced queries (V1.1)

Ordid	Prodid	Qty	Stdprice	Actprice	Totamt	Itemtot	Pctdisc
606	102130	1	3.4	3.4	3.4	3.4	0
	Sum				3.4	3.4	0
609	100861	1	45	35	45	35	22.22
	100870	5	2.8	2.5	14	12.5	10.71
	100890	1	58	50	58	50	13.79
	Sum				117	97.5	16.7
620	100860	10	35	35	350	350	0
	200376	1000	2.4	2.4	2400	2400	0
	102130	500	3.4	3.4	1700	1700	0
	Sum				4450	4450	0

Distinguishing Features

A Computations On Summaries report typically comprises several directly computed fields, as well as computed fields which are based on other computed fields. An example of the latter is shown above. The circled value is a "computation of other computations." First, the TOTAMT and ITEMTOT fields are calculated for each PRODID. Then, the TOTAMT and ITEMTOT fields are summed for each ORDID. Finally the circled PCTDISC value, which is based on these summations, is calculated.

Concepts

SQL*ReportWriter can be used to calculate summaries of summaries. In general, this can be accomplished in three ways

- Create a computed field (on the Field Settings screen) and use it as the basis for computing a summary on the Summary Screen.
- Reference any SQL*ReportWriter field in a user exit (specific on the Field Settings screen). This includes referencing fields in the SQL*ReportWriter Pre-Packaged User exits.

- Create cross-referenced queries. Across-referenced query is a query that references a field or summary via a bind parameter:

```
SELECT 100* (1 - : SUMTOT/ : SUMFULL) AVGDISC, NULL FROM DUAL
```

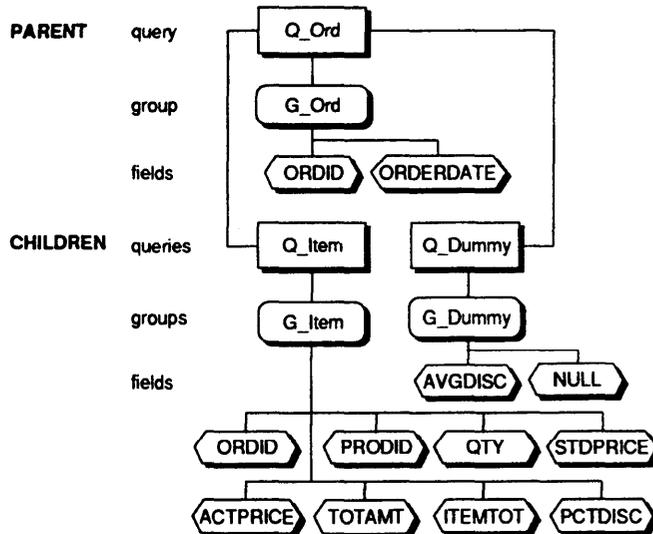
The two SQL*ReportWriter fields are references to summary fields.

Note that before a SQL*ReportWriter field maybe referenced in a query, the field must already exist. Therefore, queries that reference SQL*ReportWriter fields must appear lower in the query hierarchy than the queries fetching or computing the fields themselves.

To see a sample Computations On Summaries report that uses cross-referenced queries, open the SQL*ReportWriter example report named Adv_Comp_On_Sums_Xref. See the section called “How To Load Advanced Reports” on page 5-2 for details.

Organization

Three queries are used to build this report: one master and two detail queries. There are no manually-created groups. One of the detail queries, Q_Dummy, selects NULL from the Dual table because the other field in the G_Dummy group is skipped from the report output. (In this way, the rule for SQL*ReportWriter Version 1.1.8 and earlier—that each group must have at least one field is satisfied. Note: If you would like to prevent any of the spaces from the NULL field, and you are using SQL*ReportWriter Version 1.1.8 or later, see the report named, “Creating Groups With No Printable Fields.”)



Steps to Build this Report

- Query screen 1. Create a query, Q_Ord to fetch all order ID's associated with customer 100. Enter the following SELECT statement for Q_Ord:

```
SELECT ORDDID , ORDERDATE FROM ORD
WHERE CUSTID = 100
ORDER BY ORDDID
```

2. Enter a query, Q_Item, to fetch the product ID, quantity, Standard price, actual price, percent discount total amount before discount, and total sale amount value that are associated with each order ID. Note the use of the SQL*ReportWriter field, :orderdate, in the WHERE clause.

```
SELECT ORDDID, ITEM. PRODIG, QTY, STDPRICE,
ACTUALPRICE ACTPRICE, QTY*STDPRICE TOTAMT, ITEMTOT,
100* (1-ACTUALPRICE/STDPRICE) PCTDISC
FROM ITEM, PRICE
WHERE ITEM.PRODIG = PRICE.PRODIG
AND : ORDERDATE BETWEEN STARTDATE
AND NVL(ENDDATE(+), TRUNC(SYSDATE))
```

Join Q_Item to its parent, Q_Ord, via ORDDID.

summary screens

3. Create the following two summaries:

- A summary named SUMFULL that sums TOTAMT, has a Field Width of 10, and a print and Reset Group of G_Ord
- A summary named SUMTOT that sums ITEMTOT, has a Field Width of 10, and a Print and Reset Group of G_Ord.

Query Screen

4. Create the following query, Q_Dummy, to compute the average discount per ORDDID:

```
SELECT 100* ( 1- : SUMTOT/ : SUMFULL) AVGDISC , NULL FROM DUAL
```

Join Q_DUMMY to its parent Q_ORD. Do not specify a Common column.

Group Screen

5. Enter a 2 in the Record Spacing setting for G_Ord.

Field Screens

6. Delete the NULL Field Label.
7. Reduce the Field Width for PCTDISC to 5, and TOTAMT, ACTPRICE, and STDPRICE to 6.
8. Suppress the printing of ORDERDATE, ORDID2, and AVGDISC fields by skipping them on Field Screen Three.

Summary Screens

9. Create a summary named SUMDISC that sums AVGDISC, has a Held Width of 4, and has a Print and Reset Group of G_Ord.

Text Screen

10. Enter &SUMDISC five spaces to the right of &SUMTOT in the G_Item Footer, and insert some hyphens (-) above &SUMDISC.

Execute your report. (If desired, move the "Sum" label closer to the summary references, and type Avg just before the SUMDISC field reference in the G_Item Footer.)

Computations On Summaries using SQL (V1.1)

Ordid	Prodid	List Amount	Actual Amount	Cumulative List Amt	Cumulative Actual Amt	Cum Variance	Percent Ratio
						0.00	0%
601	200376	2.40	2.40	2.40	2.40	8.00	17%
602	100870	48.00	56.00	48.00	56.00	8.00	8%
		56.00	56.00	104.00	112.00	104.00	87%
603	100860	120.00	224.00	120.00	224.00	200.00	81%
		128.00	224.00	248.00	448.00	284.00	73%
		140.00	224.00	388.00	672.00	140.00	47%
604	100860	300.00	440.00	300.00	440.00	260.00	42%
		320.00	440.00	620.00	880.00	350.00	36%
		350.00	440.00	970.00	1,320.00	6.00	8%
	100861	78.00	84.00	78.00	84.00	6.00	8%
		84.00	84.00	162.00	168.00	0.00	0%
		90.00	84.00	252.00	252.00	12.00	3%
	100890	162.00	174.00	414.00	426.00	12.00	2%
		174.00	174.00	588.00	600.00	600.00	15%
605	100861	3,900.00	4,500.00	3,900.00	4,500.00	900.00	11%
		4,200.00	4,500.00	8,100.00	9,000.00		

Distinguishing Features

A Computations On Summaries report typically comprises several “directly” computed fields, as well as computed fields which are based on other computed fields. See the “Computations On Summaries: using cross-referenced queries,” report for an example of that type of report. This report is another example of computed fields which are based on other computed fields, but is constructed using SQL, and not SQL*ReportWriter features. This report is included in this book to illustrate alternative techniques to building Computations On Summaries reports.

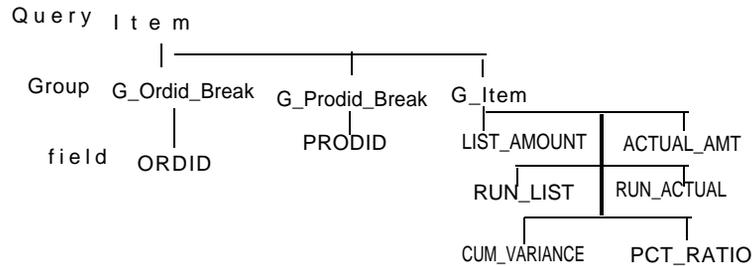
Concepts

This example report calculates percentages of cumulative columns. The Cum Budget is a running sum of the BUDGET field. The Cum variance is the absolute difference between Actual and Cum Budget. Finally, Cum Var as % of Cum Budget is the Cum Variance divided by the Cum Budget.

To view a sample Computations On Summaries report that uses SQL, open the SQL*ReportWriter example report named Adv_Comp_On_Sums_SQL. See the section called “How To Load Advanced Reports” on page 5-2 for details.

Organization

There is one query and two groups. The G_OrdId_Break and G_Prodid_Break groups were created manually.



SQL*Plus

1. Create a view called Account by entering the following SQL statement in SQL*Plus

```
CREATE VIEW ACCOUNT AS
  (SELECT ITEM. ORCID, ITEM. PRODID PRODID,
  QTY*STDPRICE LIST_AMOUNT, ITEMTOT ACTUAL_AMT
  FROM PRICE, ITEM WHERE PRICE .PRODID = ITEM. PRODID)
```

Steps to Build this Report

Query Screen

2. Create a query with a name of Item and a SELECT statement of

```
SELECT CURR.ORDID, CURR.PRODID,
  CURR.LIST_AMOUNT, CURR.ACTUAL_AMT,
  SUM (RUN.LIST_AMOUNT) RUN_LIST,
  SUM (RUN.ACTUAL_AMT) RUN_ACTUAL,
  ABS (SUM (RUN.LIST_AMOUNT) -
  SUM (RUN.ACTUAL_AMT)) CUM_VARIANCE,
  ABS ( (SUM(RUN.LIST_AMOUNT) - SUM (RUN.ACTUAL_AMT) ) *100/
  SUM (RUN.LIST_AMOUNT)) PCT_RATIO
FROM ACCOUNT CURR, ACCOUNT RUN
WHERE CURR.ORD ID = RUN.ORDID
AND RUN.PRODID <= CURR.PRODID
AND RUN.LIST_AMOUNT <= CURR.LIST_AMOUNT
AND RUN.ACTUAL_AMT <= CURR.ACTUAL_AMT
GROUP BY CURR.ORDID, CURR.PRODID, CURR.LIST_AMOUNT,
  CURR.ACTUAL_AMT
ORDER BY CURR.ORDID, CURR.PRODID, CURR.LIST_AMOUNT,
  CURR.ACTUAL_AMT
```

- Group Screens** 3. Insert a new group, G_OrdId_Break, above G_Item, and assign the new group Item. Then create another new group between the G_OrdId_Break and G_Item groups. Call it G_ProdId_Break, and assign it to the Item query.

- Field Screens** 4. Assign G_OrdId_Break to the ORDID field, and G_ProdId_Break to the PRODID field.
5. Change the following field labels

✓ **Field Labels
(modifying)**

Field Name	Label
List_Amount	List \$
Actual_Amt	Actual \$
Cum_Variance	Cum_Var
Pct_Ratio	%Ratio

6. Enter the following field widths

Field Name	Width
Ordid	3
Prodid	6
List_Amount	9
Actual_Amt	9
Run_List	10
Run_Actual	10
Cum_Variance	7
Pct_Ratio	6

7. Enter the following Display Formats

✓ **Display Formats
(altering)**

Field Name	Label
List_Amount	ZZ,ZZ9.99
Actual_Amt	ZZ, ZZ9.99
Run_List	ZZZ,ZZ9.99
Run_Actual	ZZZ,ZZ9.99
Cum_Variance	ZZZ9.99
Pct_Ratio	Z,ZZ9%

Text Screen 8. Modify the G_Item Column Heading so that it links like that below

List	Actual	Cumulative	Cumulative	Cum	Percent
Amount	Amount	List Amt	Actual Amt	Variance	Ratio
-----	-----	-----	-----	-----	-----

9. Change the G_OrdId_Break Column Heading so that it looks like that below

Ordid

10. Enter a 1 in the Spaces Before field of the G_ProdId_Break Column Heading and then modify the Text entry area so that it looks like that below:

Prodid

11. Enter a 1 in the Spaces Before field of the G_ProdId_Break Body.

Report Screen 12. Change the Report Height to 24 lines.

Execute your report.

Time Series Calculations (V1.0)

Custid	Shipdate	TOTAL	4-Month `Moving Average
100	30-JUL-86	\$3.40	\$3.40
	15-AUG-86	\$97.50	\$50.50
	01-JAN-87	\$730.00	\$730.00
	12-MAR-87	\$4,450.00	\$2,590.00
101	08-JAN-87	\$101.40	\$101.40
102	05-JUN-86	\$224.00	\$224.00
	20-JUN-86	\$56.00	\$140.00
	11-JAN-87	\$45.00	\$45.00
	05-FEB-87	\$23,940.00	\$11,992.50
	06-MAR-87	\$3,510.50	\$9,165.20
103	10-FEB-87	\$764.00	\$764.00
104	18-JUL-86	\$5.60	\$5.60
	25-JUL-86	\$35.20	\$20.40
	20-JAN-87	\$5,860.00	\$5,860.00
	04-FEB-87	\$1,260.00	\$3,560.00
105	03-MAR-87	\$46,370.00	\$46,370.00
106	30-MAY-86	\$2.40	\$2.40

Distinguishing Features

Time series calculations involve averaging values over a specified period of time. In this example, a four-month average of purchases for each customer is calculated and displayed. The techniques described can be used to produce other formats of time series calculations as well.

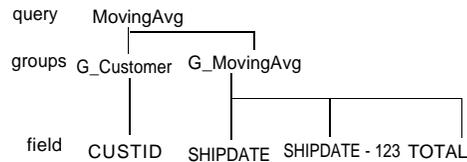
Concepts

In this Time Series Calculations report, SQL*ReportWriter computes four-month moving averages of customer purchases in the following way. A SQL statement sums the current purchase (Total) with purchases that were made by that customer in the previous four months, and then averages that sum through use of a self-pin. Therefore, if the date queried is 30-JUL-86, SQL*ReportWriter will average all purchases the customer made between 30-MAR-86 and 30-JUL-86. (The 30-MAR-86 date is determined by subtracting four months from 30-JUL-86.)

To see a sample Time Series Calculations report, open the SQL*ReportWriter example report named Adv_Time_Series_Calcs. See the section called "How To Load Advanced Reports" on page 5-2 for details.

Organization

One query is used to build this report. There is one manually-created group: G_Customer.



Steps to Build this . Report

- Query Screen 1. Create a query with a name of `MovingAvg` and a `SELECT` statement of
- ```
SELECT O. CUSPID, O. SHIPDATE, O. TOTAL, AVG (A. TOTAL)
FROM ORD O, ORD A
WHERE A. CUSTID = O. CUSTID
AND A. SHIPDATE BETWEEN O. SHIPDATE-123 AND O. SHIPDATE
GROUP BY O. CUSPID, O. SHIPDATE, O. TOTAL
```
- Group Screen 2. Insert a group above `G_MovingAvg`, name it `G_Customer`, and assign it to the `MovingAvg` query.
- Field Screen 3. Assign the `CUSTID` field to the `G_Customer` group.
4. Change the Field Label of `AVG_A_TOTAL` to 4-Month Moving Average.
5. Change the Field Width of `AVG_A_TOTAL` to 10, and enter a Display Format of `$zz, zz9.99` for both the `TOTAL` and `AVG_A_TOTAL` fields.
- Execute your report.

## Advanced Mailing Label Report (V1.1)

|                                                                 |                                                        |
|-----------------------------------------------------------------|--------------------------------------------------------|
| JOCKSPORTS<br>345 VIEWRIDGE<br>SUITE 400-G<br>BELMONT, CA 96711 | EVERY MOUNTAIN<br>574 SURRY RD.<br>CUPERTINO, CA 93301 |
| TKB SPORT SHOP<br>490 BOLI RD.<br>REDWOOD CITY, CA 94061        | K + T SPORTS<br>3476 EL PASEO<br>SANTA CLARA, CA 91003 |
| VOLLYRITE<br>9722 HAMILTON<br>BURLINGAME, CA 95133              | SHAPE UP<br>908 SEQUOIA<br>PALO ALTO, CA 94301         |
| JUST TENNIS<br>HILLVIEW MALL                                    | WOMENS SPORTS<br>VALCO VILLAGE                         |

### Distinguishing Features

This report is similar to the Introductory Mailing Label report, except that it suppresses blank address lines. The Introductory MailingLabel report prints a blank line for any null values that are retrieved from the database. For example, if you have two lines for the street (one for the street number and name, and one for the building suite number), but one of your customers does not have an address that requires the two lines, the null line will appear as a blank line in the mailing label. In this example report Null values are compressed.

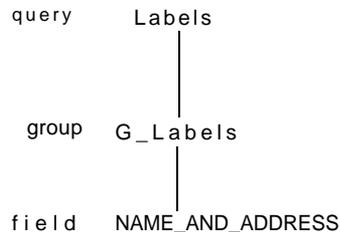
### Concepts

This report illustrates the use of Wrap fields to suppress Null values. Each line of the address is padded to 30 characters if it isn't Null. Then, all five address lines are concatenated, and the result is padded to 150 characters. Finally, this concatenated field is wrapped within a width of 30 characters. The result is mailing labels in which each address line occupies exactly 30 characters if it is non-Null. Extra blank lines appear at the end of each label to compensate for each Null address line.

To see a sample Advanced Mailing Label report, open the SQL\*ReportWriter example report named Adv\_Mailing\_Labels. See the section called "How To Load Advanced Reports" on page 5-2 for details.

## Organization

There is one query that selects all of the fields displayed in this report. There are no manually-created groups or fields.



## Steps to Build this Report

The SELECT statement shown below directly references the table called Names. As an alternative technique, you might consider creating a view based on the SELECT statement shown, and then use that view in this report. Doing so will enable you to change your underlying table(s) and column(s) in the future, without being forced to change your report.

### Query Screen

1. Create a query called Labels with the following SELECT statement:

```
SELECT RPAD (
 DECODE (CUSTNAME, ' ', ' ', (RPAD (CUSTNAME, 30))) ||
 DECODE (ADDRESS1, ' ', ' ', (RPAD(ADDRESS1, 30))) ||
 DECODE (ADDRESS2, ' ', ' ', (RPAD (ADDRESS2, 30))) ||
 DECODE (ADDRESS3, ' ', ' ', (RPAD (ADDRESS3, 30))) ||
 DECODE (ADDRESS4, ' ', ' ', (RPAD (ADDRESS4, 30)))
 , 150) NAME_AND_ADDRESS
FROM NAMES
ORDER BY CUSTNAME, ADDRESS1 , ADDRESS2, ADDRESS3, ADDRESS4
```

### Group Screens

2. set the Print Direction of the G\_Labels group to Down/Across.

### Field Screens

3. Delete the Field Label of the NAME\_AND\_ADDRESS field.
4. Change the Field Width of the NAME\_AND\_ADDRESS field to 30.

### ✓ Word-wrap Fields

5. Enter Wrap for the Align setting of the NAME\_AND\_ADDRESS field.

Execute your report.

## Invoice Report (V1.1)

|                                    |                    |                |                    |                               |
|------------------------------------|--------------------|----------------|--------------------|-------------------------------|
| <b>Summit Sporting Goods, Inc.</b> |                    |                |                    | <b>INVOICE</b>                |
| 456 Central Ave.                   |                    |                |                    |                               |
| Bigtown. CA 67890                  |                    |                |                    |                               |
|                                    |                    |                |                    | <b>Sales Order No. 025162</b> |
| Salesrep                           | ALLEN              |                |                    |                               |
| Customer                           | Every Mountain     |                |                    |                               |
|                                    | 574 SURRY RD,      |                |                    |                               |
|                                    | CUPERTINO CA 93301 |                |                    |                               |
| <u>Date</u>                        | <u>code</u>        | <u>Product</u> | <u>Description</u> | <u>Amount</u>                 |
| 18-JUL-86                          | 100871             | ACE            | TENNIS BALLS-6 P   | 5.60                          |
| 25-JUL-86                          | 101860             | SP             | TENNIS RACKET      | 24.00                         |
|                                    | 100871             | ACE            | TENNIS BALLS-6 P   | 11.20                         |
| 15-JAN-87                          | 100860             | ACE            | TENNIS RACKET I    | 3,000.00                      |
|                                    | 100861             | ACE            | TENNIS RACKET II   | 810.00                        |
|                                    | 100871             | ACE            | TENNIS BALLS-6 P   | 550.00                        |
|                                    | 101863             | SP             | JUNIOR RACKET      | 1,500.00                      |
| 22-FEB-87                          | 102130             | RH:            | "GUIDE TO TENNIS   | 340.00                        |

### Distinguishing Features

Invoice reports have several distinguishing characteristics:

- The top part of the invoice typically contains a customer name, address, and other related information. This information must all print on specific line and column positions on the pre-printed invoice form. The example report in this book named, "Printing Reports on Pre-Printed Forms" describes how to print on specific line and column positions, even if the number of lines of data being retrieved may vary at runtime.
- The middle section in most invoices contains the line-item details, including billing amounts. The number of items being printed will vary from customer to customer, and may require the invoice to span multiple pages.
- The bottom part of invoice reports contains customer billing information. Typically, this is a page total on all but the last page. On the last page, the page total is replaced with the total billing amount for that customer.

## Concepts

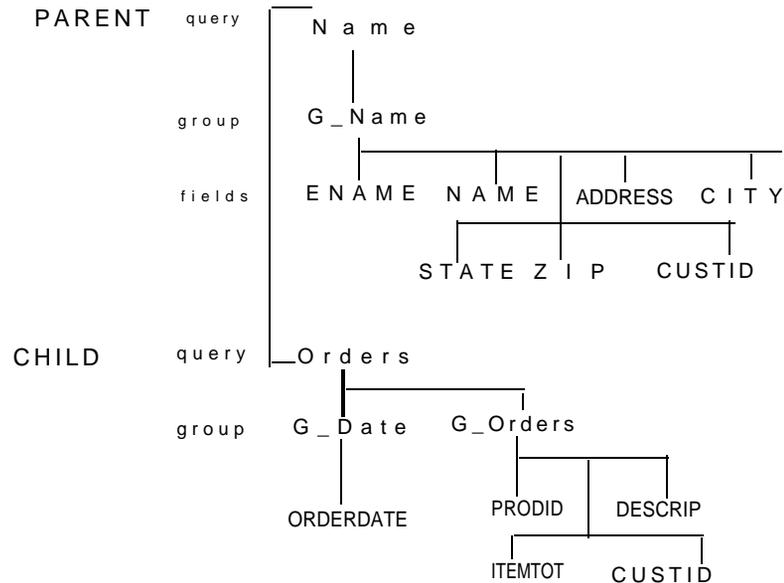
Invoice reports are Master/Detail reports with billing amounts that print conditionally. The customer name, address, and related reformation are derived from the master query (or group, if there is only one query). The line-items come from the detail query. The billingt amounts are printed in the Page Footer.

In order to print the correct billing amount, which is the page total on all pages but the last page and the invoice total on the last page, we will use the RWEIF pre-packaged user exit (if you have SQL\*ReportWriter Version 1 1.8 or earlier, substitute SRW\_IF for RWEIF). First we will compute three summaries: runningtotal, customertotal, and pagetotal. The runningtotal summary computes the running total of all line-items in the detail section. The customertotal summary computes the total billing amount for the customer. The pagetotal summary computes the page total for all line-items on a page. With the RWEIF user exit, we choose pagetotal if the runningtotal does not equal the customertotal. If it does, we choose the customertotal. Finally, we create an additional page-level summary, called printtotal, which is based on the output of the RWEIF user exit. The last step is necessary to ensure that the value of the RWEIF user exit computation is unambiguous at the page level. The printtotal summary can then be embedded in the Page Footer to print the correct billing amount.

To see a sample Invoice report, open the SQL\*ReportWriter example report named Adv\_Invoice\_Report. Seethe section called "How To Load Advanced Reports" on page 5-2 for details.

## Organization

Two queries select all of the information displayed in this report. The queries share a parent/child relationship. There is one manually-created group.



## Steps to Build this Report

### Query Screen 1. Create the following two queries **NAME**, and **ORDERS**.

- **NAME:** SELECT ENAME, NAME, ADDRESS, CITY, STATE, ZIP, CUSTID  
FROM EMP, CUSTOMER  
WHERE EMPNO = REP ID  
ORDER BY ENAME, NAME
- **ORDERS:** SELECT ORDERDATE, 1. PROD ID, DESCRIP, ITEMTOT, CUSTID  
FROM ORD O, ITEM I, PRODUCT P  
WHERE O. ORDID = I.ORDID  
AND I. PRODID = P .PRODID  
ORDER BY ORDERDATE

**Join Orders to its parent query, Name, via CUSTID.**

**Group Screens**

2. Enter a Page Break setting of Always for the G\_Name group.
3. Create a new group named G\_Date between the G\_Name and G\_Orders groups, and assign it to the Orders query.
4. Enter a 0 in the Spaces Before entry field for the G\_Date and G\_Orders groups

✓ User Exit (rweif)

Field Screens

5. Assign the ORDERDATE field to the G\_Date group.
6. Inserts field named iftotal that has a Source of

```
#rweif runningtotal >= customertotal if total
 customertotal ""
```

(If you have Version 1.1.8 or earlier of SQL\*ReportWriter, substitute SRW\_IF for RWEIF.)

7. Assign the IFTOTAL field to the G\_Orders group, change its Field Width to 10, and change its DataType to NUM.
8. On the previous screen, Field Screen One, delete all of the field labels.
9. On Field Screen Two, enter a Display Format of zz, zz9. 99 for the ITEMTOT field.
10. Enter the following Relative Position settings

| Field Name | Relative Position |
|------------|-------------------|
| NAME       | Below             |
| ADDRESS    | Below             |
| CITY       | Below             |

11. Enter Variable in the Align entry field for NAME and CITY.
12. Skip the CUSTID, IFTOTAL, and CUSTID2 fields (place an x in their Skip entry fields).

## ✓ Summary Fields

### Summary Screens

13. Enter the following summaries

- PAGETOTAL that sums ITEMTOT, that has a Field Width of 10, a Display Format of \$ZZ,ZZ9.99, a Print Group of G\_Name, and Reset Group of page.
- PRINTTOTAL that has a function of Last of the IF TOTAL field, has a Field Width of 10, a Display Format of \$zz,zz9.99, a Print Group of G\_Name, and a Reset Group of page.
- RUNNINGTOTAL that computes a running sum of ITEMTOT, has a Field Width of 10, a Print Group of G\_Orders, and Reset Group of G\_Name.
- CUSTOMERTOTAL that sums ITEMTOT, has a Field Width of 10, a Print Group of G\_Orders, and Reset Group of G\_Name.

### Text Screen

14. Go to the text object of Page with the Type of Footer, enter a 3 in the Lines Before entry field and 50 in the Spaces Before entry field, and then enter the following in the Text entry area:

```
Page Total = & PAGE TOTAL
Total: &PRINTTOTAL
```

15. Go to the text object of G\_Name with the Type of Body, enter a 2 in the Lines Before entry field and 8 in the Spaces Before entry field, and then enter the following in the Text entry area:

```
&ENAME &CR
&NAME &CR
&ADDRESS &CR
&CITY, &STATE &ZIP &CR
```

16. Go to the text object of G\_Orders with the Type of Body, enter Margin for the Relative Position, and a 10 in the Spaces Before entry field.
17. Go to the text object of G\_Date with the Type of Body, enter Margin for the Relative Position, and a 7 in the Lines Before entry field.

### Report Screen

Change the Page Height from 66 to 24 lines per page, and change the Top Margin to 0.

Execute your report.

## Check Printing Report (V1.1)

| <input type="radio"/> SUMMIT SPORTING GOODS<br>123 MAIN STREET<br>anytown, ca 12345<br><br><input type="radio"/> 40736<br>30-NOV-1990<br><br><input type="radio"/> PAY TO THE ORDER OF <u>VOLLYRITE</u> \$ [REDACTED]<br><input type="radio"/> XXXX AND XX/100 DOLLARS<br><br><input type="radio"/> 001077:12207000.: AUTHORIZED <sup>YYY</sup> SIGNATURE | <input type="radio"/> SUMMIT SPORTING GOODS<br>123 MAIN STREET<br>anytown, ca 12345<br><br><input type="radio"/> 40737<br>30-NOV-1990<br><br><input type="radio"/> PAY TO THE ORDER OF <u>CANCELLED</u> \$ [REDACTED]<br><input type="radio"/> XXXX AND XX/100 DOLLARS<br><br><input type="radio"/> 001077:12207000.: AUTHORIZED <sup>YYY</sup> SIGNATURE | <input type="radio"/> INVOICE<br>Sales Order No. 025162<br><br><input type="radio"/> <table border="1"> <thead> <tr> <th>Ordid</th> <th>Shipdate</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>602</td> <td>20-JUN-86</td> <td>56.00</td> </tr> <tr> <td>603</td> <td>05-JUN-86</td> <td>224.00</td> </tr> <tr> <td>611</td> <td>11-JAN-87</td> <td>45.00</td> </tr> <tr> <td>614</td> <td>05-FEB-87</td> <td>23,940.00</td> </tr> </tbody> </table><br><input type="radio"/> INVOICE<br>Sales Order NO. 025163<br><br><input type="radio"/> <table border="1"> <thead> <tr> <th>Ordid</th> <th>Shipdate</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>618</td> <td>06-MAR-87</td> <td>3,510.50</td> </tr> </tbody> </table> | Ordid | Shipdate | Total | 602 | 20-JUN-86 | 56.00 | 603 | 05-JUN-86 | 224.00 | 611 | 11-JAN-87 | 45.00 | 614 | 05-FEB-87 | 23,940.00 | Ordid | Shipdate | Total | 618 | 06-MAR-87 | 3,510.50 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|----------|-------|-----|-----------|-------|-----|-----------|--------|-----|-----------|-------|-----|-----------|-----------|-------|----------|-------|-----|-----------|----------|
| Ordid                                                                                                                                                                                                                                                                                                                                                     | Shipdate                                                                                                                                                                                                                                                                                                                                                  | Total                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |          |       |     |           |       |     |           |        |     |           |       |     |           |           |       |          |       |     |           |          |
| 602                                                                                                                                                                                                                                                                                                                                                       | 20-JUN-86                                                                                                                                                                                                                                                                                                                                                 | 56.00                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |          |       |     |           |       |     |           |        |     |           |       |     |           |           |       |          |       |     |           |          |
| 603                                                                                                                                                                                                                                                                                                                                                       | 05-JUN-86                                                                                                                                                                                                                                                                                                                                                 | 224.00                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |     |           |       |     |           |        |     |           |       |     |           |           |       |          |       |     |           |          |
| 611                                                                                                                                                                                                                                                                                                                                                       | 11-JAN-87                                                                                                                                                                                                                                                                                                                                                 | 45.00                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |          |       |     |           |       |     |           |        |     |           |       |     |           |           |       |          |       |     |           |          |
| 614                                                                                                                                                                                                                                                                                                                                                       | 05-FEB-87                                                                                                                                                                                                                                                                                                                                                 | 23,940.00                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |       |          |       |     |           |       |     |           |        |     |           |       |     |           |           |       |          |       |     |           |          |
| Ordid                                                                                                                                                                                                                                                                                                                                                     | Shipdate                                                                                                                                                                                                                                                                                                                                                  | Total                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |          |       |     |           |       |     |           |        |     |           |       |     |           |           |       |          |       |     |           |          |
| 618                                                                                                                                                                                                                                                                                                                                                       | 06-MAR-87                                                                                                                                                                                                                                                                                                                                                 | 3,510.50                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |          |       |     |           |       |     |           |        |     |           |       |     |           |           |       |          |       |     |           |          |

### Distinguishing Features

This Check Printing report is built so that the check appears on the left, and its related stub on the right. It could, however, be built so that the check appears above the stub. There is one interesting aspect to building any check printing report: canceling checks for stubs that require more than one "page."

The company name is printed on the check for the first page of the stub, and "CANCELLED" is printed on all checks thereafter for stubs that are continued onto more than one "page."

### Concepts

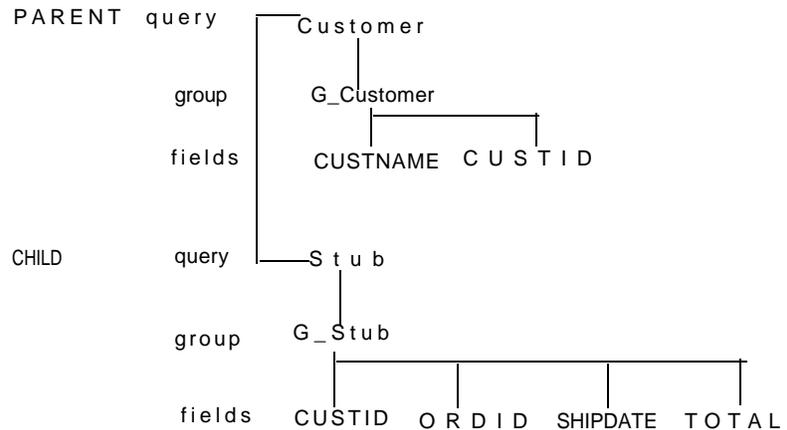
The challenge in building this example report is to cancel the invalid checks. To do so, we take advantage of SQL\*ReportWriter's ability to overwrite boilerplate text (or even fields) with other fields.

In this report there are two text objects that are defined to appear in the same location: one contains the field, "&CUSTNAME," and the other contains, "CANCELLED." The first text object is such that it prints only on the first "page" of a multi-page check stub, while the second prints on all pages. We set the Relative Position of the first text object to Margin, which makes it overwrite the second object. Thus, page one will contain the name of the customer (right next to the :Pay to the Order of" line in the check), while all subsequent pages will contain the word, "CANCELLED."

To see a sample Check Printing report, open the SQL\*ReportWriter example report named Adv\_Check\_Report. See the section called "How To Load Advanced Reports" on page 5-2 for details.

## Organization

Two queries select all of the information displayed in this report. The queries share a parent/child relationship. There are no manually-Created groups.



## Steps to Build this Report

### Query Screen 1. Create the following two queries: Customer and Stub.

- Customer: `SELECT RPAD (NAME, 30) CUST_NAME, CUSTID  
FROM CUSTOMER ORDER BY CUSTID`
- Stub: `SELECT CUSTID, ORDID, SHIPDATE, TOTAL  
FROM ORD ORDER BY ORD ID`

Join Stub to its parent query, Customer. Join the queries on the CUSTID column.

**Group Screens**

- 2. Set the Page Break for G\_Customer to Always, and set Spa-Before for G\_Stub to 10.

**Field Screens**

- 3. Delete the Field Label for CUST\_NAME.
- 4. Enter a Display Format of zzz,zz9.99 for the TOTAL field.
- 5. Skip the CUSTID and CUSTID2 fields (Place an x in the Skip entry field for the two fields).

**Text Screen**

- 6. Edit the G\_Customer text object of Type Body to look like the text shown below:

&DATE

CANCELLED  
xxxx and xx/10 0

YYY

- 7. Set the Relative Position of G\_Stub Header to Margin, and then edit the text to look like the text shown below

&DATE

&CUST\_NAME  
xxxx and xx/100

YYY

- 8. Set the Relative Position of the G\_Stub Column Heading to Right, set Spaces &Before to 8, and enter G\_Customer for its Frequency.

**Report Screen**

Change the Report Height to 10. .

Execute your report To print checks above the stubs, move the G\_Stub Column Heading and Body text object to Panel 2 after the report is created.

## Spelling Out Cash Amounts On Checks (V1.1)

|                                                                                          |                       |
|------------------------------------------------------------------------------------------|-----------------------|
| Name EVERY MOUNTAIN<br>SEVEN THOUSAND ONE HUNDRED SIXTY AND 80/100                       | Sum Total \$7,160.80  |
| Name JOCKSPORTS<br>FIVE THOUSAND TWO HUNDRED EIGHTY AND 90/100                           | Sum Total \$5,280.90  |
| Name JUST TENNIS<br>SEVEN HUNDRED SIXTY-FOUR AND 0/100                                   | Sum Total \$764.00    |
| Name K + T SPORTS<br>FORTY-SIX THOUSAND THREE HUNDRED SEVENTY AND 0/100                  | Sum Total \$46,370.00 |
| Name NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER<br>SIX THOUSAND FOUR HUNDRED AND 0/100 | Sum Total \$6,400.00  |
| Name SHAPE UP<br>NINE THOUSAND TWENTY-FOUR AND 40/100                                    | Sum Total \$9,024.40  |
| Name TKB SPORT SHOP<br>ONE HUNDRED ONE AND 40/100                                        | Sum Total \$101.40    |

### Distinguishing Features

There are various ways you can spell out the cash amount of a check. The method used in this report uses the ORACLE DECODE function and a lookup table to perform the conversion. This report can spell out amounts from zero to 999,999.99. (This report is being explained separately from the Check Printing report for simplicity.)

### Concepts

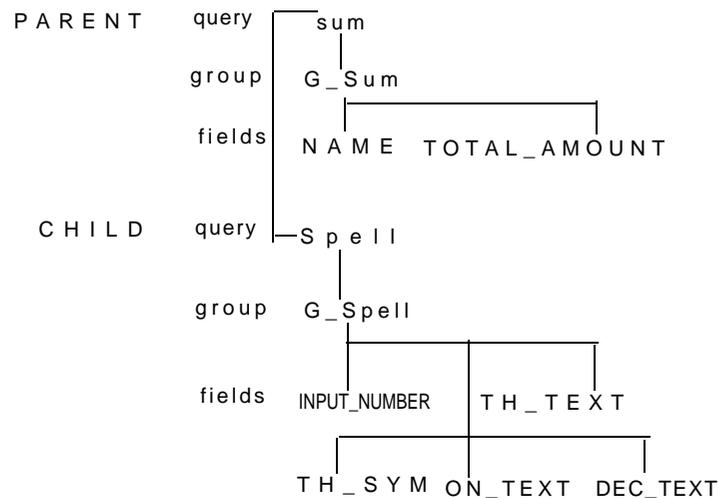
The queries in this report assume that the numbers to be spelled out are stored in the column named TOTAL, in the table called Oral. The spelled-out number text will be returned in the Thou, Ones, and Dec columns. A lookup table is used to fetch the spelled-out number.

You can create the lookup table by executing ADVANCED.SQL, a SQL script that is shipped with SQL\*ReportWriter.

To see a sample report spelling out cash amounts on checks, open the SQL\*ReportWriter example report named Adv\_Spell\_Cash\_Amts. See the section called "How To Load Advanced Reports" on page 5-2 for details.

## Organization

Two queries are used to build this report: Sum and Spell. The queries share a parent/child relationship.



- SQL\*Plus** 1. While in the demo directory, run ADVANCED.SQL to create a table named LOOKUP. (If you can't locate the file, check your Installation and User's Guide for more information.)

## Steps to Build this Report

- Query Screen** 1. Create a query named `Sum` with the following **SELECT** statement:

```
SELECT NAME, SUM (TOTAL) ,
FLOOR (SUM (TOTAL) /1000) THOU,
TRUNC (SUM (TOTAL) - TRUNC (SUM (TOTAL) , -3) , 0) ONES,
(SUM (TOTAL) - TRUNC (SUM (TOTAL) , 0)) *100 DEC
FROM CUSTOMER, ORD
WHERE CUSTOMER . CUSTID = ORD . CUSTID
GROUP BY NAME
ORDER BY NAME
```

**Create a query named Spell with the following SELECT statement**

```
SELECT DECODE (: THOU, 0, '', A. NUMBERTEXT I I ' THOUSAND ') I I
DECODE (:ONES, 0,
DECODE (:THOU,0,'ZEROAND ' II :DEC II '/100','AND' II
:DEC II '/100'),B.NUMBERTEXT11 'AND' II :DEC II
'/100') SPELLED_VALFROMLOOKUPA, LOOKUP B
WHERE A.NUMBERVALUE = :THOU AND B.NUMBERVALUE= :ONES
```

Join Spell to its Parent Query, sum, without specifying any columns on which to join.

**Group Screens**

Enter a Relative Position of `Below` for `G_Spell`, and a 1 in the spacing (Record) entry for `G_Sum`. Then on Group Screen Three, enter `Left` for the Label Position of both groups.

**Field Screens**

4. Delete the Label for `SPELLED_VAL`.
5. On Field Screen Two, change the Field Width of `SUM_TOTAL` to 12, `SPELLED_VAL` to 70, and then enter a Display Format of `$zzz,zz9.99` for `SUM_TOTAL`.

Skip the `THOU`, `ONES`, and `DEC` fields.

Execute your report. You may want to modify the report appearance.

## Matrix With Zeros for Null Values (V1.0)

|    | ANALYST | CLERK | MANAGER | PRESIDENT | SALESMAN |
|----|---------|-------|---------|-----------|----------|
| 10 | 0       | 1     | 1       | 1         | 0        |
| 20 | 2       | 2     | 1       | 0         | 0        |
| 30 | 0       | 1     | 1       | 0         | 4        |

### Distinguishing Features

By default, matrix reports display a space for null (blank) values. In this example report, zeros replace null values.

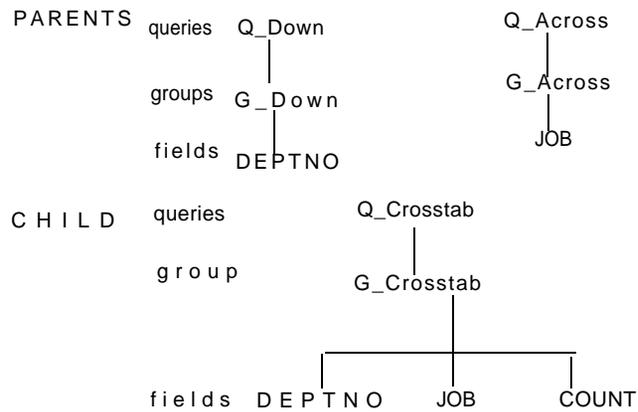
### Concepts

By default, matrix reports display blanks for Null crosstab values. This happens whenever the Q Crosstab query (see below) does not retrieve any records. In this example report, zeros are displayed. This is accomplished by a SQL statement that counts the records that are fetched. Thus, if no records are fetched, a value of zero appears where the blank would have otherwise appeared.

To see a sample Matrix report with zeros for null values, open the SQL\*ReportWriter example report named Adv\_Matrix\_With\_Zeros. See the section called "How To Load Advanced Reports" on page 5-2 for details.

## Organization

Three queries select the data for this report Q Down, Q Across, and Q Crosstab. AU of the fields are owned by their default groups.



## Steps to Build this Report

### Query Screen

1. Create the following three queries Q\_Down, Q\_Across, and Q\_Crosstab.

- Q\_Down: SELECT DISTINCT DEPTNO FROM EMP
- Q\_Across: SELECT DISTINCT JOB FROM EMP
- Q\_Crosstab: SELECT DISTINCT A. DEPTNO, B. JOB, O COUNT  
FROM EMP A, EMP B  
WHERE (A. DEPTNO, B. JOB) NOT IN  
(SELECT DISTINCT DEPTNO, JOB FROM EMP)  
UNION  
SELECT DEPTNO, JOB, COUNT(EMPNO) FROM EMP  
GROUP BY DEPTNO, JOB

**Join Crosstab to its parent queries, Down and Across, with the DEPTNO and JOB fields.**

### Group Screens

2. Assign a Print Direction of Down, Across, and crosstab for the Down, Across, and Crosstab queries. Then, specify the groups to be Matrix Groups.

**Field Screens**

3. Enter `Left` for the Align setting of the DEPTNO field. (Because the default Justification for NUM fields is Right, changing it to Left will put more room between the DEPTNO values and the first column of the matrix.)
4. Delete all field Labels.
5. Change the Field Width of COUNT to 4.
6. Skip the second occurrence of the DEPTNO and JOB fields.

Execute your report. Add summaries or computed fields as desired.

## Matrix Break Report (V1.0)

|           |        |        |        |        |        |        |                   |          |
|-----------|--------|--------|--------|--------|--------|--------|-------------------|----------|
| Customer: | 106    |        |        |        |        |        | "Total Purchases: | \$ 9,024 |
| SHAPE UP  |        |        |        |        |        |        |                   |          |
| Product   | APR 86 | MAY 86 | JUN 86 | JUL 86 | AUG 86 | SEP 86 |                   |          |
| -----     | -----  | -----  | -----  | -----  | -----  | -----  |                   |          |
| 100860    |        |        | 440    |        |        |        |                   |          |
| 100861    |        |        | 84     | 4,500  |        |        |                   |          |
| 100870    |        |        |        | 1,400  |        |        |                   |          |
| 100890    |        |        | 174    | 290    |        |        |                   |          |
| 101860    |        |        |        | 1,200  |        |        |                   |          |
| 101863    |        |        |        | 900    |        |        |                   |          |
| 102130    |        |        |        | 34     |        |        |                   |          |
| 200376    |        | 2      |        |        |        |        |                   |          |
| Customer: | 106    |        |        |        |        |        | Total Purchases:  | \$ 280   |
| VOLLYRITE |        |        |        |        |        |        |                   |          |
| Product   | APR 86 | MAY 86 | JUN 86 | JUL 86 | AUG 86 | SEP 86 |                   |          |

### Distinguishing Features

A Matrix Break report is a combination of Matrix and Break report formats. Essentially, a matrix-format report is printed for each master group value. This report shows specific information about each customer and their associated products.

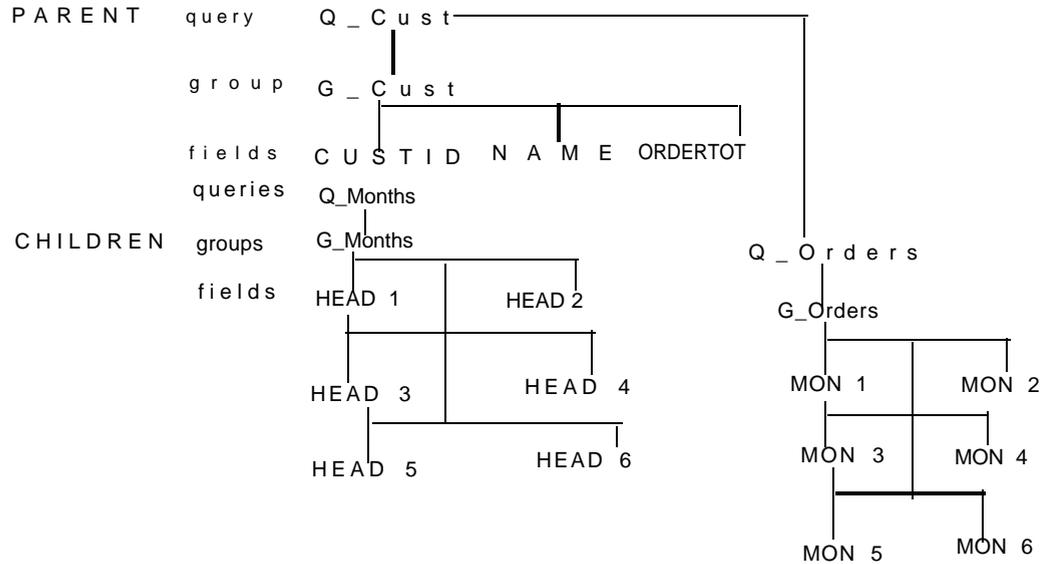
### Concepts

To create this report, you must create three queries. The first specifies the customer information (master), the second specifies the cross-tabulation information about products and months (detail), and the third specifies the names of the months. Note that the matrix portion of the report is constructed "manually"; that is, in SQL.

To see a sample Matrix Break report, open the SQL\*ReportWriter example report named Adv\_Matrix\_Break. See the section called "How To Load Advanced Reports" on page 5-2 for details.

## Organization

Three queries are used to build this report they share a master/detail/detail relationship. There are no manually-created groups.



## Steps to Build this Report

### Query Screen 1. Create a Query with a name of Q\_Cust and a SELECT statement of

```

SELECT CUSTOMER . CUSTID , NAME, SUM (TOTAL) ORDERTOT
FROM CUSTOMER, ORD
WHERE CUSTOMER . CUSTID = ORD . CUSTID
AND ORDERDATE BETWEEN
 ADD_MONTHS (: CURDATE, -5) AND : CURDATE
GROUP BY CUSTOMER. CUSTID , NAME
ORDER BY SUM (TOTAL) DESC

```

**2. Create a Query with a name of Q\_Months and a SELECT statement of**

```
SELECT CUSTID,
TO_CHAR (ADD_MONTHS (: CURDATE, -5) , ' MON YY') HEAD1,
TO_CHAR (ADD_MONTHS (: CURDATE, -4) , ' MON YY') HEAD2,
TO_CHAR(ADD_MONTHS (: CURDATE, -3), 'MON YY') HEAD3,
TO_CHAR(ADD_MONTHS (: CURDATE, -2), 'MON YY') HEAD4,
TO_CHAR (ADD_MONTHS (: CURDATE, -1), ' MON YY ') HEAD5,
TO_CHAR (TO_DATE (:CURDATE)t 'MON YY') HEAD6,
FROM CUSTOMER
```

**Enter Q\_cuSt in the parent Query 1 entry field, and then join the Q\_Months query to Q\_Cust with the CUSTID column.**

**3. Create a Query with a name of Q\_Orders and a SELECT statement of**

```
SELECT CUSTID, PRODID,
SUM(DECODE(TO_CHAR(ORDERDATE~ 'MON YY'), TO_CHAR(
ADD_MONTHS (:CURDATE, -5), 'MON YY'), ITEMTOT, 0)) MON1,
SUM(DECODE (TO_CHAR(ORDERDATE, 'MON YY'), TO_CHAR(
ADD_MONTHS (:CURDATE , -4), 'MON YY'), ITEMTOT, 0))MON2,
SUM(DECODE (TO_CHAR(ORDERDATE, 'MON YY'), TO_CHAR(
ADD_MONTHS (:CURDATE , -3), 'MON YY'), ITEMTOT, 0))MON3,
SUM(DECODE (TO_CHAR(ORDERDATE, 'MON YY'), TO_CHAR(
ADD_MONTHS (:CURDATE , -2), 'MON YY'), ITEMTOT, 0))MON4,
SUM(DECODE (TO_CHAR(ORDERDATE, 'MON YY'), TO_CHAR(
ADD_MONTHS (:CURDATE , -1.), 'MON YY'), ITEMTOT/ 0)) MON5,
SUM(DECODE (TO_CHAR(ORDERDATEt 'MON YY'), TO_CHAR(
TO_DATE (:CURDATE), 'MON YY'), ITEMTOT, 0)) MON6
FROM ORD, ITEM
WHERE ORD.ORDID = ITEM.ORDID
GROUP BY CUSTID, PRODID
ORDER BY CUSTID, PRODID
```

**Enter Q\_cmt in the Parent Query 1 entry field, and then join the Q\_Orders query to Q\_Cust with the CUSTID Column.**

## Group Screens

4. Move to Group Screen Two and enter the following Relative Position and spacing entries  
G\_Cust: Record Spacing: 3  
Field Spacing: 3  
G\_Months: Relative Position: Below  
Lines Before 2  
Spaces Before 0  
G\_Orders: Relative position: Below  
Spaces Before 0
5. Go to Group Screen Three and enter `Left` for the Label position of G\_Cust.

## Field Screens

6. Move to Field Screen One and change the Label of Custid to `Customer:`, change the Ordertot Label to `Total Purchases:`, and delete all of the other field labels.
7. Move to Field Screen Two and change the following Field Widths  
ORDERTOT to 9, HEAD1 through 6 to 6, PRODID to 6, MON1 through 6 to 6.

Enter the following Display Formats for the fields listed:

| Field Name | Display Format |
|------------|----------------|
| ORDERTOT   | \$BZZZ, ZZ9    |
| MON 1      | ZZ,ZZZ         |
| MON2       | ZZ,ZZZ         |
| MON3       | ZZ,ZZZ         |
| MON4       | ZZ,ZZZ         |
| MON 5      | ZZ,ZZZ         |
| MON6       | ZZ,ZZZ         |

9. Enter a Relative Position of Below for the NAME field, and a Relative Position of Right for the ORDERTOT field.
10. Move to Field Screen Three and Skip the CUSTID2 and CUSTID3 fields by placing an x in their corresponding Skip entry areas.

**Text Screen**

Move to the text object where the Object is G\_Months and the Type is Body. Change the Text entry area so that it looks like that below

```
PRODUCT &HEAD1 &HEAD2 &HEAD3 &HEAD4 &HEAD5 &HEAD6

```

**Parameter Screen**

Create the following parameter on Parameter Screen One

| <b>Parameter Name</b> | <b>Data Type</b> | <b>Width</b> | <b>Default Value</b> | <b>Label</b> |
|-----------------------|------------------|--------------|----------------------|--------------|
| CURDATE               | DATE             | 9            | 01-SEP-86            | Curdate      |

**Execute your report.**

## Ranking Report (V1.1)

| Top 3 Customers: |                 |
|------------------|-----------------|
| Customer Name    | Total Purchases |
| K + T SPORTS     | \$46,370.00     |
| VOLLYRITE        | \$27,775.50     |
| SHAPE UP         | \$9,024.40      |

| Top 75% of Sales: |                 |
|-------------------|-----------------|
| Customer Name     | Total Purchases |
| K + T SPORTS      | \$46,370.00     |
| VOLLYRITE         | \$27,775.50     |

### Distinguishing Features

This report ranks data in two different ways: by count, and by percentage. The upper section displays the names and total purchases of the top three customers; the lower section displays the names and total purchases of those customers who constitute 75% of all sales. The ranking criteria can be set by the user at runtime, or default to previously specified values.

This explanation is split into two parts. The first part tells you how to build the ranking-by-count section. The second part discusses building the ranking-by-percent section on top of the first section.

### Concepts

This example report ranks the data by comparing it to a user-specified bind parameter. In this example, the parameter value is 3. That is, the top three customers should be displayed.

To rank the data, you create the following SQL\*ReportWriter object:

- a bind parameter for the cutoff point. In this example, the cutoff default is 3 (for the top 3 customers).
- a computed field which keeps a running count of the total purchases

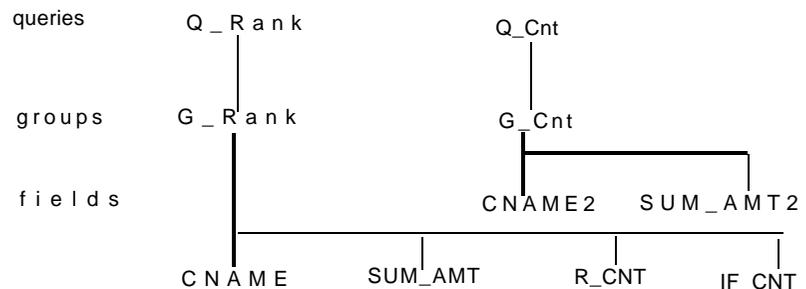
- a field with a Source using the RWEIF pre-packaged user exit to compare the running count of the customer totals to the cutoff parameter. (If you have SQL\*ReportWriter Version 1.1.8 or earlier, substitute SRW\_IF for RWEIF).
- a summary to set the cutoff point in the data.

These objects determine the values that are needed to query the desired results. Finally, you create a second query to select the records above the cutoff point.

To see a sample Ranking report, open the SQL\*ReportWriter example report named Adv\_Ranking. See the section called “How To Load Advanced Reports” on page 5-2 for details.

## Organization

Two queries and two groups are used to build this section of the report. There are no manually-created groups.



## Steps to Build this Report

**Query Screen**    **1. Create a query named Q\_Rank with the following SELECT statement:**

```

SELECT CUSTNAME CNAME, SUM (AMOUNT) SUM_AMT FROM SALES
GROUP BY CUSTNAME
ORDER BY SUM (AMOUNT) DESC

```

**Parameter Screens**    **2. Create a parameter named CUTOFF\_CNT with a Data Type of NUM, a Field Width of 1, and a Default Value of 3.**

✓ User Exit (RWEIF)

**Field Screens**

3. Below SUM\_AMT, create a computed field named R\_CNT that has a Source SUM\_AMT, a group of G\_Rank, a Field Width of 10, and a Function of R\_Count.
4. Below R\_CNT, create a field named IF\_CNT that has a Source of 

```
#rweif r_cnt <= cutoff_cnt if_cnt sum_amt ""
```

 a group of G\_Rank, a DataType of NUM, and a Field Width of 10.
5. On Field Screen Three skip CNAME, SUM\_AMT, R\_CNT, and IF\_CNT.

**Summary Screens**

6. Create a summary named MINSUM\_CNT that computes the minimum Of Field IF\_CNT and has a Print and Reset Group setting of REPORT.

**Query Screen**

7. Create a second query called Q\_Cnt, with the following SELECT Statement  

```
SELECT CUSTNAME CNAME2 , SUM (AMOUNT) SUM_AMT2 FROM SALES
GROUP BY CUSTNAME HAVING SUM (AMOUNT) >= :MINSUM_CNT
ORDER BY SUM (AMOUNT) DESC
```

**Field Screens**

8. Change the Label of CNAME2 to Customer Name, SUM\_AMT2 to Total Purchases, and change their Field Width and Display Format to those below:

| <b>Field Name</b> | <b>Field Width</b> | <b>Display Format</b> |
|-------------------|--------------------|-----------------------|
| CNAME2            | 35                 |                       |
| SUM_AMT2          | 15                 | \$ZZZ,ZZZ,ZZ9.99      |

**Text Screens**

Move to the G\_Cnt Header text object and enter the following text:

Top &CUTOFF\_CNT Customers:

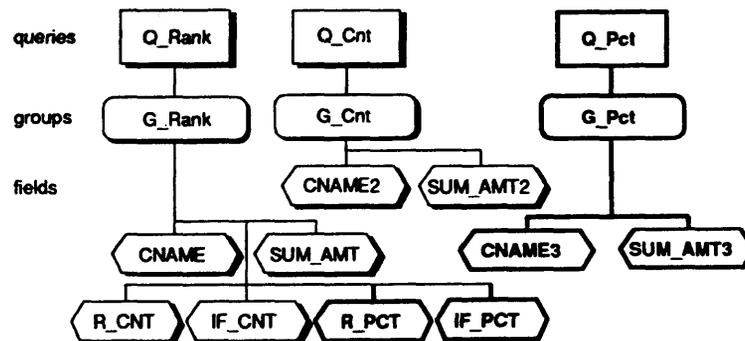
Move to the G\_Cnt Column Heading text object and enter a 1 for the Lines Before setting.

Execute your report.

Change the Default Value for CUTOFF\_CNT to 5 and run the report again. You should now see the top five customers.

## Related Concepts

Another way to rank customers is by a percentage of total sales. The principles are the same you specify a cutoff percentage parameter, and compare the running total percent of sales to the cutoff. You can build the percentage ranking portion of the report by adding it to the report you just built. The objects you will add to your report are highlighted below:



## Steps to Build this Report

- |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter Screens | 1. Create a parameter named <code>CUTOFF_PCT</code> with a DataType of <code>NUM</code> , a Field Width of 2, and a Default Value of 75.                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Field Screens     | 2. Below <code>IF_CNT</code> , create a computed field named <code>R_PCT</code> with a Source of <code>SUM_AMT</code> , a group of <code>G_Rank</code> , a Field Width of 10, and a Function of <code>R%_Total</code> .<br>3. After <code>R_PCT</code> , create a field named <code>IF_PCT</code> with a Source of <pre>#rweif r_pct &lt;= cutoff_pct if_pct sum_amt ""</pre> a group of <code>G_Rank</code> , a DataType of <code>NUM</code> , and a Field Width of 10.<br>4. Move to Field Screen Three and Skip fields <code>R_PCT</code> and <code>IF_PCT</code> . |
| Summary Screens   | 5. Create a summary named <code>MINSUM_PCT</code> that computes the minimum of the <code>IF_PCT</code> field and has a Print and Reset Group setting of <code>REPORT</code> .                                                                                                                                                                                                                                                                                                                                                                                          |

**Query Screen**

6. Create a third query named Q\_Pct with the following SELECT Statement:

```
SELECT CUSTNAME CNAME3 , SUM (AMOUNT) SUM_AMT3 FROM SALES
GROUP BY CUSTNAME HAVING SUM(AMOUNT) >= :MINSUM_PCT
ORDER BY SUM(AMOUNT) DESC
```

**Group Screen**

Enter a Relative Position of Below for G\_PCT, a 3 for Lines Before and a 0 for Spaces Before.

**Field Screens**

Change the Label for CNAME3 to customer Name, and SUM\_AMT3 to Total Purchases.

On Field Screen Two change the following settings

| Field Name | Field Width | Display Format   |
|------------|-------------|------------------|
| CNAME3     | 35          |                  |
| SUM_AMT3   | 15          | \$ZZZ,ZZZ,ZZ9.99 |

**Text Screens**

Move to the G\_Pct Header text object and enter the following text:

Top &CUTOFF\_PCT% of Sales:

Move to the G\_Pct Column Heading text object and enter a 1 for the Lines Before setting.

Execute your report.

11. Change the Default Value of CUTOFF\_PCT to 4 and run the report again. You should see the top four percent of customers.

## Changeable Number Of Records per Column Report (V1.1)

| Ename  | Sal  |        |             |        |      |
|--------|------|--------|-------------|--------|------|
| SMITH  | 800  | BLAKE  | <b>2850</b> | ADAMS  | 1100 |
| ALLEN  | 1600 | CLARK  | 2450        | JAMES  | 950  |
| WARD   | 1250 | SCOTT  | 3000        | FORD   | 3000 |
| JONES  | 2975 | KING   | 5000        | MILLER | 1300 |
| MARTIN | 1250 | TURNER | 1500        |        |      |

### Distinguishing Features

This report is similar to a Down/Across or Across/Down report, except that at runtime a user-specified number is used to format the report. The number tells SQL\*ReportWriter how many rows should appear in each column. In this example, the user entered a 5, specifying that 5 records should appear in each column.

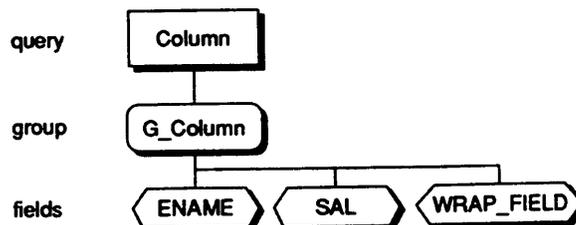
### Concepts

A parameter to calculate how many records SQL\*ReportWriter should place in each column, and the Align setting of wrap, are used to produce this report. Using the report settings, SQL\*ReportWriter calculates how many rows of data will fit across one line, and assigns the result to a field. This field is then given an Align setting of Wrap, which causes the field's contents to be formatted onto multiple lines.

To see a sample report that allows a changeable number of records per column, open the SQL\*ReportWriter example report named Adv\_Chg\_Recs\_Per\_Col. See the section called "How To Load Advanced Reports" on page 5-2 for details.

### Organization

One query and one group are used to build this report. There are no manually-created groups.



## Steps to Build this Report

**Report Screen** 1. Change the Page Height from 66 to 24 lines per page.

**Query Screen** 2. Create a query named `Column` with the following SELECT statement

```
SELECT ENAME , SAL,
 RPAD(' ',
 DECODE
 (MOD (ROWNUM, : COLUMN_LENGTH) ,0,1,0)
 * (<Page Height> - <Top Margin> - <Bottom Margin>
 -<Column Heading Height>- :COLUMN_LENGTH) * 2,
 ' ') WRAP_FIELD
FROM EMP
```

**Note:** In this query, <Page Height>, <Top Margin>, <Bottom Margin>, and <Column Heading Height> are static values in your report. In this example, your values should be as follows

| <u>Setting</u>        | <u>Value</u> |
|-----------------------|--------------|
| Page Height           | 24           |
| Top Margin            | 2            |
| Bottom Margin         | 2            |
| Column Heading Height | 2            |

**Group Screens** 3. For `G_Column` enter a Print Direction of Down/Across.

**Field Screens** 4. Delete the Label for `WRAP_FIELD`, change its Field Width to 1, and enter wrap for Align setting.

**Parameter Screen** 5. Change the Data Type of `COLUMN_LENGTH` to `NUM`, change its Width to 3, and give it a Default Value of 5.

Execute your report.

6. Execute the report again, and when the Run-time Parameter Form appears enter a value other than 5 for the `COLUMN_LENGTH` parameter. (Try 2, then 10.)

## Conditional Printing (V1.1)

| Deptno | Dname      | Ename       | Sal  |
|--------|------------|-------------|------|
| 10     | ACCOUNTING | CLARK       | 2450 |
|        |            | KING        | 5000 |
|        |            | MILLER      | 1300 |
|        |            | Sum Salary: | 8750 |
|        |            | Min Salary: | 1300 |
| 20     | RESEARCH   | ADAMS       | 1100 |
|        |            | FORD        | 3000 |
|        |            | JONES       | 2975 |
|        |            | SCOTT       | 3000 |
|        |            | SMITH       | 800  |
|        |            | Min Salary: | 800  |

### Distinguishing Features

Conditional Printing reports display database values and/or text information only when certain criteria are met. You use this type of report when you want to determine which database values or text objects should be displayed in the report. In this example, a text object that contains the salary sum computation is displayed for department 10 only.

The conditional printing technique can also be used to produce form letters with varying contents.

### Concepts

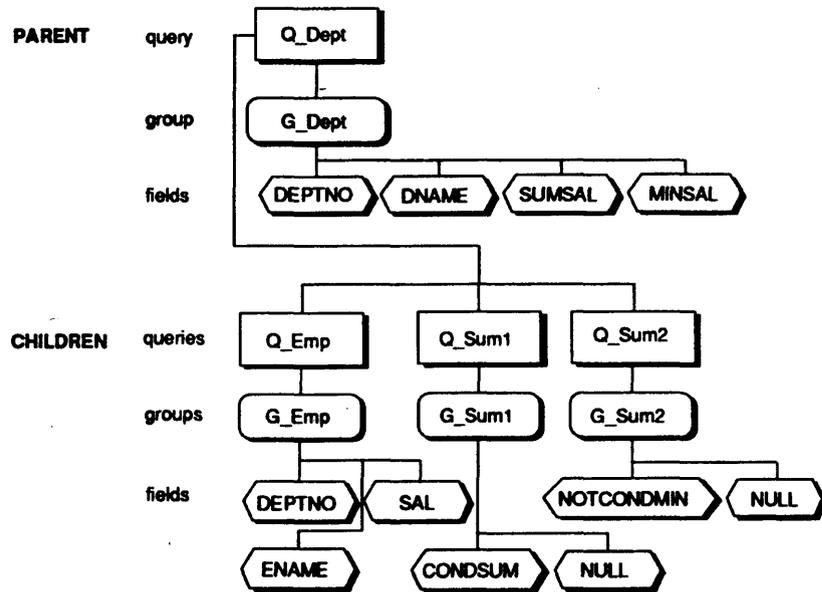
To compute and display the sum of salaries for department 10 only, you set 10 to be the default department for which you will print the salary sum. One of your queries then uses a bind parameter which allows you to enter any department number at runtime, and during execution the RWECPF user exit (if you have SQL\*ReportWriter Version 1.1.8 or earlier, substitute SRW\_COPYFIELD for RWECPF) copies that department number to a new field. If the department number in the new field equals the default, the salary summary for that department (in this case, department 10) prints.

If you change the department number at runtime, the salary summary prints for the department you selected.

To see a sample Conditional Printing report, open the SQL\*ReportWriter example report named Adv\_Cond\_Printing. See the section called "How To Load Advanced Reports" on page 5-2 for details.

## Organization

Four queries are used to build this report. One query is a parent of the other three queries. There are no manually-created groups.



## Steps to Build this Report

**Query Screen** 1. Create the following four queries: Q\_Dept, Q\_Emp, Q\_Sum1, and Q\_Sum2:

- Q\_Dept: SELECT DEPTNO, DNAME FROM DEPT  
ORDER BY DEPTNO
- Q\_Emp : SELECT DEPTNO, ENAME, SAL FROM EMP ORDER BY ENAME

Join Q\_Emp to its parent, Q\_Dept, and specify the DEPTNO column for the pin.

- Q\_Sum1: SELECT NULL FROM DUAL  
WHERE :DEPTNO= : DEPTNO\_TO\_COMPUTE

**Join Q\_Sum2 to its parent, Q\_Dept, without specifying any Columns on which to join.**

- Q\_Sum2: SELECT NULL FROM DUAL

Join Q\_Sum2 to its parent, Q\_Dept, without specifying any columns on which to pin.

### Group Screens

2. Enter a Relative Position of Below for G\_Sum1 and G\_Sum2, and a Record Spacing of 2 for G\_Dept.

Now enter the following for Spaces Before

| Group Name | Spaces Before |
|------------|---------------|
| G_Dept     | 0             |
| G_Emp      | 2             |
| G_Sum1     | 0             |
| G_Sum2     | 0             |

3. On Group Screen Three, enter a Label Position of Left for G\_Sum1 and G\_Sum2.

### Field Screens

4. Below DNAME, create the following two fields: SUMSAL and MINSAL, with a Source of SAL and a Group of G\_Dept.

5. Enter the following Functions and Reset Groups:

| Field Name | Function | Reset Group |
|------------|----------|-------------|
| SUMSAL     | Sum      | G_Dept      |
| MINSAL     | Min      | G_Dept      |

### ✓ User Exit (rwecpf)

6. Below NULL, create a third new field named CONDSUM. Enter as its source

```
#rwecpf sumsal condsum
```

and assign it to Group G\_Sum1. Move to Field Screen Two and enter a DataType of NUM and a Field Width of 10 for CONDSUM.

7. Below NULL2, create a final new field named `NOTCONDMIN`. Enter as its Source

```
#rwecpf minsal notcondmin
```

and assign it to Group `G_Sum2`. Move to Field Screen Two and enter a DataType of NUM and a Field Width of 10 for NOTCONDMIN.

8. Move to Field Screen Three and Skip SUMSAL, MINSAL, DEPTN02, NULL, and NULL2.

### Text Screens

9. Move to the `G_Sum1` Body text object and enter the following in the Text entry area:

```

```

```
Sum Salary &condsum
```

Move to the `G_Sum2` Body text object and enter the following in the Text entry area:

```

```

```
Min Salary ¬condmin
```

### Parameter Screens

10. Change the Data Type of `DEPTNO_TO_COMPUTE` to NUM, the Width to 2, and enter a Default Value of 10.

Execute your report. Notice that the "Salary Sum" field appears for department 10 only.

11. Execute the report again, changing `DEPTNO_TO_COMPUTE` to 20, or 30, on the Run-time Parameter Form. Notice that "Salary Sum" is now displayed only for the department you specify.

CHAPTER

# 6

## INTEGRATING SQL\*REPORTWRITER

**T**his chapter describes the concepts and building steps for the integration methods listed below:

- passing parameters from a SQL\*Forms-based application to a SQL\*ReportWriter report
- calling a report from a 3GL program using the SQL\*ReportWriter Call Interface.

## Integrating SQL\*ReportWriter and SQL\*Forms

Parameters enable you to pass information from SQL\*Forms to SQL\*ReportWriter. You can use parameters to control many different aspects of your report, including report destination, printer type, number of copies, and to specify what data to use in the report. Any number of parameter values can be passed.

### Steps to Build this Report

To pass parameters from SQL\*Forms to SQL\*ReportWriter, you need to do the following:

- build your report with SQL\*ReportWriter, and specify default values for your runtime parameters
- build a form using SQL\*Forms (Design) and, if needed, specify defaults for the field values that are passed to SQL\*ReportWriter
- determine a trigger with which to invoke SQL\*ReportWriter and pass the parameters. Any trigger type can be used.
- use the HOST packaged procedure in your trigger to pass control to SQL\*ReportWriter, and the desired parameters. After executing the report, control will automatically return to SQL\*Forms. Refer to the following charts for assistance.
- use SQL\*Forms (Run Form) and fire the trigger.

### Trigger Syntax Using PL/SQL-based Triggers

The syntax you should use for your PL/SQL-based trigger will vary, based on the number of parameters you want to pass to SQL\*ReportWriter. Use the following table for assistance:

|                                             | Small Number of Parameter Values                                                                    | Large Number of Parameter Values                                                                                         |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>Parameter Values Are Known and Fixed</b> | <pre>HOST (' runrep report= myreport userid=myname/ mypassword param=myparam' ) ;</pre>             | <pre>HOST (' runrep cmdfile=A. B' ) ;</pre>                                                                              |
| <b>Parameter Values Can Vary</b>            | <pre>HOST (' runrep myreport user id='   :my ID    'param1='   :Var1    'param2='    :var2 );</pre> | <p><i>Insert parameter values into a temporary table and select from the table in your SQL*Report Writer report.</i></p> |

**Trigger Syntax without PL/SQL-based Triggers**

To determine the syntax you should use for a non-PL/SQL trigger, refer to the following table:

|                                             | Small Number of Parameter Values                                                                            | Large Number of Parameter Values                                                                                |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Parameter Values Are Known and Fixed</b> | #EXEMACRO HOST ' runrep<br>report=myreport<br>userid=myname/mypassword<br>param=myparam' ;                  | #EXEMACRO HOST ' runrep<br>myreport cmdfile=A. B' ;                                                             |
| <b>Parameter Values Can Vary</b>            | #EXEMACRO HOST ' runrep myreport<br>userid=' I I :my ID I I 'param1='<br>I I :var1 I I 'param2=' I I :var2; | Insert parameter values into a temporary<br>table and select from the table in your<br>SQL*ReportWriter report. |

The number of parameter values is important because some operating systems limit the number of parameters you can pass. For example, within VMS you cannot pass more than eight parameters if you use the syntax in the left column.

**Integrating a Form and Report for Summit Sporting Goods**

Let's say that you wanted to pass some parameters from a form (Figure 6-1) to an invoice report. The parameters that you would likely pass are the username/password, and the customer name. (You would pass the customer name so that SQL\*ReportWriter creates and outputs the invoice for that customer only.)

**FIGURE 6-1**  
Sample SQL\*Forms Application

```

===== CUSTOMER =====

CUSTID 100
NAME JOCKSPORTS
ADDRESS 345 VIEWRIDGE
CITY BELMONT ZIP 96711
STATE CA PHONE 98-6609
AREA 415 CREDITLIMIT 5000
REPID 7844
COMMENTS Very friendly people to work with -- sales rep likes to be c

Count: #1 <Replace>

```

The syntax you would use to run the invoice report would be:

```
HOST ('runrep customer_invoice_report userid=:user_var II
'paramform=no batch=yes custname=' II :cust_name, NO_SCREEN);
```

(Don't forget to include the semi-colon!)

The NO\_SCREEN argument in the trigger instructs SQL\*Forms to neither clear the screen nor prompt the operator before returning to SQL\*Forms after the report has been run. This allows you to invoke SQL\*ReportWriter invisibly to the user.

Note: When using the SQL\*Forms HOST command and specifying NO\_SCREEN, make sure that the executable included in the HOST command does not cause any screen I/O. If it does, your display and key mapping could be altered when returning to SQL\*Forms. When using the NO\_SCREEN option, make sure to specify BATCH=YES in your RUNREP command.

## Calling SQL\*ReportWriter Using the Call Interface

You can invoke SQL\*ReportWriter from your 3GL programs using the SQL\*ReportWriter Call Interface. The Call Interface enables you to call SQL\*ReportWriter functions that execute SQLREP, RUNREP, GENREP, DUMPREP, or LOADREP. Each procedure takes one parameter a valid SQL\*ReportWriter command line string containing any number of SQL\*ReportWriter arguments. All SQL\*ReportWriter Call Interface function calls return a positive number if an error is encountered, or a zero if no error is detected.

For more information on using the SQL\*ReportWriter Call Interface, refer to Appendix D of the SQL\*ReportWriter Reference Manual V1.1.

### Examples

The following C programs call the RWCRUN Call Interface routine to invoke RUNREP.

- NEWRUN1.PC logs on to the RDBMS and passes the name of the report, the destination type, and a flag indicating that the report should be run in batch mode. The program then commits any data definitions or manipulations the report may have performed.

(Note: When a SQL\*ReportWriter Call Interface function does not receive a logon string in the command line string parameter, it does not issue a COMMIT as it does when a logon string is passed. Therefore, as shown in this case, the calling function must perform the commit.)

- NEWRUN2.C passes the name of the report, the ORACLE logon string the destination type, and a flag indicating that the report should be run in batch mode.

NEWRUN1.PC and NEWRUN2.C are listed on the following pages.

## NEWRUN1.PC

```
#include <stdio. h>
#define SQLCA_STORAGE_CLASS extern
EXEC SQL INCLUDE SQIIC.A;

main ()
{
 EXEC SQL BEGIN DECLARE SECTION ;
 VARCHAR uid[240] ;
 VARCHAR pwd[30] ;
 EXEC SQL END DECLARE SECTION;

 long ret;

 /*****
 /*Case1: Run RUNREP; calling program logs onto RDBMS and */
 /* commits work. */
 *****/

 /* Log on to RDBMS */
 strcpy ((char *) uid.arr, "scott");
 uid.len = (unsigned short) 5;
 strcpy ((char *) pwd.arr, "tiger");
 pwd.len = (unsigned short) 5;
 EXEC SQL WHENEVER SQL ERROR goto sqlerr;
 EXEC SQL CONNECT :uid IDENTIFIED BY :pwd;

 /* CallRUNREP -- don't pass logon string */
 ret = rwcrun ("myreport1 destype=file batch=yes") ;
 if (ret) goto runerr;

 /*Commit work, if any DDL or DML done in the report*/
 EXEC SQL COMMIT WORK RELEASE;

 exit(0);

sqlerr:
 printf ("Oracle error -- %n ", sqlca .sqlcode);
 exit(0) ;
runerr:
 printf ("Error running RUNREP .");
 exit(0);
}
```

## NEWRUN2.C

```
#include <stdio. h>

main ()
{
 long ret;

 /******
 /* Case2: RunRUNREP; have SQL*ReportWriter log on to RDBMS */
 /******

 /* callRUNREP with logon string */
 ret = rwcrun("myreport2 scott/tigerdestype=file batch=yes") ;
 if (ret) goto runerr;

 exit,(0);

runerr:
 printf ("Error running RUNREP .");
 exit(0);
}
```

NEWRUN1.PC should be pre-compiled and compiled, and NEWRUN2.C should decompiled. To link the 3GL programs with SQL\*ReportWriter, you need to edit the link script that is shipped with SQL\*ReportWriter. In particular, you need to replace the entry routine (for RUNREP, the entry routine is RUNMAI) with the new C entry routine. You may also wish to rename the executable to a name other than RUNREP.

The sample VMS link script shown below displays the modifications needed to include the object file, NEWRUN1 (the modified lines are highlighted in bold). NEWRUN2 could replace the call to NEWRUN1 shown in this sample link script.

**Link Script (modified  
LRUNREP.COM)**

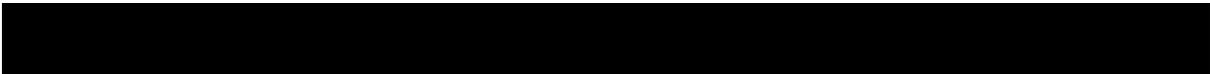
```

$! LINK REPORTWRITER CALL INTERFACE
$ ASSIGN/NOLOG SYS$ORACLE ORA_RDBMS
$ ASSIGN/NOLOG SYS$ORACLE ORA_UTIL
$ define/nolog SRW_OLB ORA_SQLREPORTWRITER_OLB
$ LOUtl : == @SYS$ORACLE : LOUtl
$ OKANS := "123"
$ XITOBJF = "SRW_OLB: rwxtb"
$ DONEXIT
$ NLSLIB = ""
$ OCILIB = "OCI"
$ V56LIB= ""
$ ORACLE = "SRW_OLB:evadrc, "
$ if "ORA_VSN" .eqs. "6"then go to LINKIT
$ NLSLIB = "ORA_SQLREPORTWRITER_OLB:lmslib/1,"
$ OCILIB="ocilib"
$ V56LIB= "ORA_UTIL:v56lib/1/i= (lstrlcmp,lmemlcmp) ,"
$ ORACLE= "ORA_RDBMS:oracle/1, "
$!

$ LINKIT:
$ loutl newrun-
newrun1 , -
SRW_OLB:rwexec, -
SRW_OLB:rwstl, -
SRW_OLB:aaaaf1, -
SRW_OLB:rwepfo, -
SRW_OLB:rweprs, -
'XITOBJF', -
SRW_OLB:lrunrep/opt, -
SRW_OLB:rwint/1/i=aucvar, -
SRW_OLB:rwgen/1/i=(aucrun, aucrlm), -
SRW_OLB:ally/1, -
'NLSLIB'-
'ORACLE'-
'V56LIB' -
ORA_RDBMS:'OCILIB'/1/i=ocicee,-
ORA_UTIL:sqllib/1-
newrun'pi' 'p2' 'p3'

```

When you are linking with the Call Interface, you need to include all SQL\*ReportWriter libraries and object files. Note that RWXTB (line 7) is used when you have no user exit object files; if you have user exit object files, you need to include them and IAPXTB in the link script, and remove RWXTB from the link script.



## GLOSSARY

**action menu** The Action Menu lists a set of operations to perform on the selected report. The options are New, Open, Copy, Rename, Drop, Execute, Generate, and Quit.

**across/down** A print direction that first prints each record of the group to the right of the previous record until the right margin of the page is reached. Printing then returns to just below the left-most record on the same page, and continues across.

**alert box** A pop-up window that asks for confirmation of an action that would otherwise cause the changes that were made to be lost.

**alias** A temporary name used in a SQL statement to reference a table or a column expression.

**align** (Field Screens) The values of fields may be aligned to the left, right, or center of the defined width of a field. Fields can also be wrapped, on word boundaries, onto multiple lines; or be of variable width (i.e., have trailing spaces truncated).

**ancestor group** A group that is above another group, either directly or indirectly.

**argument** An expression within the parentheses of a function, supplying a value for the function to operate on. For example, on the command line, `KEYWORD=VALUE` is an argument

**attribute** See setting.

**body** The text that is repeated for every record of a group.

**boilerplate text** The text that you enter on the Text Screen. The text will appear exactly as it appears on the Text Screen. Note: If you use the Wrap or Variable options of the Align setting on the Field Screens, your boilerplate text will word-wrap. See the SQL\*ReportWriter Reference Manual for details.

**break report** A report that consists of more than one group, where the groups are placed side by side. This is also known as an outline report.

**browse mode** Allows you to view report output on the screen page by page.

**call interface** An interface that allows RUNREP and/or SQLREP to be linked with other Oracle products, such as SQL\*Menu, or a user-written program.

**child query** Also called a detail query. When defining a master/detail report, by defining multiple tables, the child query retrieves details information for values retrieved by the parent, or master, query.

**CMDFILE** A command line argument that allows you to specify a file that contains a set of arguments for SQLREP or RUNREP.

**column expression** An expression in a SELECT statement that defines the data that is retrieved from the database (see Field). May be a column name or a valid SQL expression referencing a column name.

**column header** A type of text. By default, a column header contains the labels for the fields in the associated group.

**command line** Operating system command line. SQL\*ReportWriter can be invoked from the operating system command line using a number of parameters See SQLREP and . RUNREP in the SQL\*ReportWriter Reference Manual.

**comment** (Report Screen) The comment setting provides a space to enter descriptive information about a report.

**compile** See Generate.

**computed field** (Field Screens) Used to perform report-time calculations on data retrieved from the database. These calculations augment the kinds of calculations that can be done directly with a SELECT statement.

**control break** See break report.

**crosstab** A print direction that prints each record of a group as the cell in a matrix. See Matrix Report.

**cursor** The lighted rectangle or bar on the terminal's screen which shows where the next character entered will appear.

**Data Type** (Field, Parameter, Summary Screens) Specifies the type of data contained in a field. Maybe CHAR, NUMBER, DATE, or PRT.

**date** One of ORACLE's datatypes. A date column may contain a date and time between January 1,4712 BC to December 31,4712 AD.

**default** A value that is used if no alternative is specified.

**DESFORMAT** A command line argument that allows you to specify the file that describes the Characteristics Of the printer named in DESNAME.

**DESNAME** A command line argument that allows you to specify the file or printer to which the report output will be written.

**DESTYPE** A command line argument that allows you to specify the type of output device that will receive report output.

**dialog box** A window asking the user to enter some piece of information necessary to complete an operation. Dialog boxes are used extensively in the Action Menu.

**display format** (Field and Summary Screens) Defines the appearance of the value of a field.

**down/across** A print direction that first prints each record of the group below the previous record until the bottom of the page is reached. Then, records will be printed to the right of the top-most record on the same page.

**DUMPREP** Command which unloads a report definition into an ASCII export file. This file can then be moved to another computer. This command is used in conjunction with the LOADREP command.

**execute** (Action Menu) Reduces the report output.

**field** A container for the data retrieved by a particular column expression in a query. Define settings to control how the value in a field is displayed.

**field label** (Field Screens) Text that appears by default on the report output to describe a column of fields in a report.

**field name** (Field Screens) The name of a field. You must name every field for reference in other parts of the report definition. (See SQL Naming Standards in the SQL\*ReportWriter Reference Manual for details.)

**field spacing** (Group Screens) Specifies the default number of blank spaces between all fields in the specified group for Down and Down/Across groups. Specifies the default number of blank lines between all fields in the specified group for Across and Across/Down groups.

**field width** (Field Screens) Specifies the width for each field.

**fields across** (Group Screens) Specifies the maximum number of fields in a group that will be placed on a single line of a panel by default for Down and Down/Across groups. Specifies the maximum number of fields that will be placed in a single column for Across and Across/Down groups.

**footer** A text type that prints after a group (group footer), at the bottom of each page (page footer), or at the end of the Report (report footer). In matrix reports, this is called a sub-footer.

**frequency** (Text Screen) Defines how often the column heading will be printed.

**FROM** Required clause of the SELECT statement that identifies the tables from which data is selected.

**function** (Field and Summary Screens) Computes the value of a field or summary.

**generate** (Action Menu) Creates a runfile enabling reports to be run via RUNREP.

**GENREP** A command which generates a runfile even in the absence of a database. This command is used in conjunction with RUNREP, which runs a report.

**group** Groups define sections and/or subsections of reports, and are also used for subtotals and totals.

**GROUP BY** Optional clause of a SELECT statement which identifies the level of summarization to be performed before fetching data from the database.

**group list** The list of groups found on the Groups screens.

**group name** (Group Screens) The name of a group. Groups are named so they can be referenced by other objects.

**group's query** The query associated with the group on the Group Settings screen.

**header** A text type that precedes the group (group header, prints at the top of a page (page header), or at the beginning of the report (report header).

**highlight** (Field, Group, and Text Screens) On the Field Screens, individual fields can be highlighted in the label area. Also on this screen, conditional highlighting can occur using user exits On the Group Screen, fields in a group can be highlighted. On the Text Screen, Mark and Highlight enable the highlighting of anything: parts of fields or text, to entire reports See Printer Control Codes in the SQL\*ReportWriter Reference Manual).

**index** An alphabetical listing of the topics available in the HELP system.

**Join** Combining data from two (or more) tables in a single SELECT statement.

**justification** (Text Screen) The alignment of text, either right, left, or center.

**label** (highlight) (Field Screens) Specifies the highlighting style for the label of a field.

**label position** (Group Screens) The position in which to print the field labels in the Body or Column Heading.

**left margin** (Report Screens) The number of space that SQL\*ReportWriter skips from the left of every page before outputting data.

**lines before** (Group, Field, Text Screens) The number of vertical lines to insert before printing an object.

**list of values** A list of the valid values in the current context. In some cases, the LIST shows existing names which cannot be used.

**LOADREP** Executable command which loads an ASCII export file into an ORACLE database. This is used in conjunction with the DUMPREP command.

**LOGFILE** A command line argument which allows you to identify the file used by the interactive print commands [Print Page], [Print Screen], and [Print Report].

**margin** See Left Margin and Right Margin.

**matching columns** (Query Screen) Information needed to link the data in two or more queries together. See Parent Query.

**matrix** (Group Screens) Indicates that a group is part of a matrix report.

**matrix report** Across-tabulation of data from three groups. The values in one group form the rows, the values from the second group form the columns, and the values from the third group form the cells.

**message line** The line at the bottom of the screen where SQL\*ReportWriter displays hints and other messages, including error messages.

**multi-panel** (Group Screens) Specifies whether all fields in a group should remain on the *same* panel.

**name** (Report Screen) Specifies the name of the device to send report output to.

**object** An item in a report, for example, groups and fields, pages, and the report as a whole.

**ORDER BY** Clause used in a SELECT statement to specify the order in which the results of a query are to be displayed.

**output** Results of a report definition after it is executed. Output can be displayed on a screen, stored in a file, or printed on a hard copy.

**overflow** Overflow occurs when all of the records of a group do not fit on a single page.

**page** The area on which report objects are placed.

**page break** (Group Screens) Controls whether or not to generate a page break.

**page height** (Report Screen) Specifies the height of the physical page in lines.

**page width** (Report Screen) The width of the physical page in character spaces.

**Panel** A panel is a chunk of text. Each text object may consist of one or more panels, each of which appear on a separate page.

**panel number** (Text Screen) The number of the panel on which to print text.

**PARAMFORM** A command line argument which allows you to display or suppress the Run-time Parameter Form when you execute a report.

**parent query** (Query Screen) When defining a master/detail report, by joining multiple tables, the parent query retrieves master information for values retrieved by the child query.

**PRINTDEF** Executable command that is used to create a new printer definition.

**print direction** (Group Screens) The direction in which each record of the group prints relative to the prior record.

**print group** (Summary and Field Screens) The Print Group determines how often the summary prints.

**printer control codes** Codes that system administrators can define which enable users to highlight text and fields with any print capabilities of the printer.

**product\_user\_profile** A view which enables security at the report level when SQL\*ReportWriter is installed on the ORACLE transaction processing option.

**query** A SQL SELECT statement that retrieves information from one or more tables or views.

**query Name** (Query and Group Screens) Name that identifies queries to be referenced in group settings or as the parent of another query.

**quit** (Action Menu) Option from the List of Actions that terminates a session and returns the user to the operating system.

**read consistency** The ability to query the data throughout a database as of a single point in time. This ensures that all the data in a report is internally consistent.

**READONLY** A command line argument that allows you to request read Consistent without locking for a multiple query report by locking the tables in SHARE MODE.

**record** In a group, a record represents a row from a query. Adding a group manually creates a mock record for each distinct set of values of the fields in a group.

**record spacing** (Group Screens) The number of blank lines or spaces between the records in a group.

**relative position** (Group, Field, and Text Screens) The position of a group or field relative to the previous group or field. With regard to text, the location of text relative to the previous object.

**rename** (Action Menu) Used to change the name of a report.

**repeat** (Field Screens) This causes the field to appear on all panels of the report.

**repeat on page overflow** (Text Screen) Specifies whether a text object repeats on subsequent pages if the data overflows.

**REPORT** A command line argument that allows you to specify the report to execute.

**reset group** (Field and Summary Screens) The group at which the value of a function is to be reset to 0.

**right margin** (Report Screens) The number of spaces that SQL\*ReportWriter skips from the right of every page before outputting data. Note that if the group has a prior sibling of its own, the right boundary of that sibling will be used for the margin of the texts.

**row** One set of fields in a table.

**run** See execute.

**runfile** A compact representation of the report definition which is used at runtime to control the output of the report.

**running summary** Every summary function has a running version which returns cumulative values of the function between reset points.

**RUNREP** Operating system command that executes previously defined reports

**SELECT** statement (Query Screen) A SQL statement used to select rows and columns from one or more database tables.

**setting** An area in which you specify a value for a SQL\*ReportWriter object characteristic.

**sibling** A group that is below another group, either directly or indirectly.

**skip** (Field Screens, Parameter Screens) Specifies whether SQL\*ReportWriter will output or suppress the indicated field in the location implied by the field settings. If enabled, the field will appear only when manually added to text objects. Also used to suppress parameters from appearing on the Run-time Parameter Form=

**source column** (Field ~) A COI~ name, expression, or alias appearing in a SELECT statement.

**source query** The query that produced the source column for the current field.

**spaces before** (Group, Field and Text Screens) The number of horizontal column positions to insert before printing an object.

**SQL** Standard interface for storing and retrieving information in a relational database. SQL is an acronym for "Structured Query Language."

**SQLREP** Operating system command that invokes SQL\*ReportWriter and allows you to define and execute reports.

**status** (Text Screen) Non-enterable field which indicates whether the current text is "Default" or "Edited."

**subtotal** See summary.

**summary** summaries, or summary fields, are used to compute subtotals, grand totals, running totals, and other summarizations of the data in a report.

**summary field** Field containing data derived by SQL\*ReportWriter using one of the summary functions. Summary fields are displayed by default in the group footer.

**summary name** (Summary Screens) The name given to the field being summarized for reference in other part of the report definition.

**system variable** Variables that are provided by SQL\*ReportWriter (i.e., &DATE, &NUM\_PAGES, &PAGE\_NUM).

**system-owned tables** SQL\*ReportWriter tables that are not owned by users: the users share these tables with other users.

**TERM** A command line argument that allows you to specify a file describing the terminal on which you are using SQLREP or RUNREP.

**text** (Text Screen) Texts contain field reference, labels, and boilerplate text, that appear at various points in your report. See Text Type.

**Text Type** (text screen) Each group, and the report as a whole, has several text objects associated with it Each text object has a type that defines where it will appear in the report output. Valid group types are header, footer, body, column heading and subfoot. Valid report text objects are title page, trailer page, report header and footer, and page header and footer.

**title page** If a report title page is used, it is the first page of the report and contains only text from the title page text object on the Text Screen. Otherwise, it is nonexistent no blank page is printed.

**toggle** Switch that can be turned on or off to control the behavior of report output or the definition process. For example, you can toggle between insert and overtype mode when entering data.

**top margin** (Report Screen) The number of lines that SQL\*ReportWriter skips from the top of every page before outputting any text or data, including headings.

**trailer page** Text object that appears at the end of a report.

**type** (Report Screen) Specifies the type of device to send report output to.

**USERID** A command line argument which allows you to specify your ORACLE username and password with an optional SQL\*Net address.

**user exit** Away in which to pass arguments from SQL\*ReportWriter to another computer program and, if desired, pass new or modified arguments to SQL\*ReportWriter.

**user-owned** tables SQL\*ReportWriter tables that are owned by users: the users do not Share these tables with other users

**variable** The fitting, or wrapping of a field's contents within a defined text object width (the width is fixed; the depth is dependent upon the number of characters the field contains). The variable field's data Will not overwrite other text.

**view** A virtual table whose rows do not actually exist in the database. A virtual table is based on a table that is physically stored in the database.

**window mode** Allows you to view report output on the screen a line at a time. This is used for output that is less than one page but greater than one screen.

**width** (Held, and Text Screens) The width of a field, page, or panel. Width is measured in characters a space is a character.

**word wrap** The fitting, or wrapping of a field's contents within a defined field width (the width is fixed; the depth is dependent upon the number of characters the field contains). The wrapped field's data may overwrite other text.

## INDEX

### !

- &DATE 2-26
- &NUM\_PAGES 2-26
- &PAGE 2-26
- &SQL (feature)
  - how to use 4-34
  - reports that use it 3-2

### A

- Across reports with control breaks
  - brief description of 3-17
  - concepts of 5-35
  - distinguishing features 5-35
  - explanation of 5-35
  - organization of 5-36
  - sample report output 3-17
  - steps to build 5-36 -5-37
- Action menu
  - using 1-6
- Advanced mailing label report
  - brief description of 3-35
  - concepts of 5-50
  - distinguishing features 5-50
  - explanation of 5-50
  - organization of 5-51
  - sample report output 3-35
  - steps to build 5-51
- Advanced mailing labels (feature)
  - reports that use them 3-2

- Aggregating data (feature)
  - reports that use it 3-2
- Aggregating data within ranges report
  - brief description of 3-16
  - Concepts of 5-28
  - distinguishing features 5-28
  - explanation of 5-28
  - organization of 5-29
  - sample report output 3-16
  - Steps to build 5-29
- Align setting
  - See Variable-length fields
  - See Word-wrapped fields

### B

- Body 2-24
- Boilerplate text (feature)
  - how to build 4-28
  - reports that use it 3-2
- Break report
  - brief description of 3-13
  - concepts of 4-7
  - distinguishing features 4-6
  - explanation of 4-6
  - organization of 4-7
  - sample report output 3-13
  - steps to build 4-8- 4-9
  - See also Master/Detail report
- Break, control (feature)
  - See Control break (feature)
- Browse mode 1-18

## C

- Call interface
  - example 6-5
  - using 6-5
- Carriage returns, embedding (feature)
  - how to build 4-29
- Changeable number of records per column
  - brief description 3-31
  - concepts of 5-76
  - distinguishing features 5-76
  - explanation of 5-76
  - organization of 5-76
  - sample report output 3-31
  - steps to build this report 5-77
- Check printing report
  - brief description of 3-37
  - concepts of 5-57
  - distinguishing features 5-57
  - explanation of 5-57
  - organization of 5-58
  - sample report output 3-37
  - steps to build 5-58- 5-59
- Check printing report-spelling out amounts
  - See Spelling out cash amounts report
- Column heading 2-23
- Column headings, deleting (feature)
  - reports that have deleted headings 3-2
- Computations On Summaries report: using SQL
  - brief description of 3-10
  - concepts of 5-44
  - distinguishing features 5-44
  - explanation of 5-44
  - organization of 5-45
  - sample report output 3-10
  - steps to build 545 - 5-47
- Computations On Summaries report: using cross-referenced queries
  - brief description of 3-9
  - concepts of 5-40
  - distinguishing features 5-40
  - explanation of 5-40
  - organization of 5-41
  - sample report output 3-9
  - steps to build 5-42 - 5-43
- Computed fields (feature)
  - how to build 4-22
  - reports that use them 3-2
- Computed fields report
  - brief description of 3-8
  - concepts of 4-21
  - distinguishing features 4-20
  - explanation of 4-20
  - sample report output 3-8
  - steps to build 421 - 4-23
- Computed ruining totals (feature)
  - reports that use them 3-2
- Concepts 2-1
- Conditional column headings (feature)
  - reports that use them 3-2
- Conditional highlighting (feature)
  - reports that use it 3-2
- Conditional highlighting report
  - brief description of 3-33
  - concepts of 5-10
  - distinguishing features 5-10
  - explanation of 5-10
  - organization of 5-11
  - sample report output 3-33
  - steps to build this report 5-11, 5-13
- Conditional printing (feature)
  - reports that perform it 3-2
- Conditional printing report
  - brief description of 3-32
  - concepts of 5-78
  - distinguishing features 5-78
  - explanation of 5-78
  - organization of 5-79
  - sample report output 3-32
  - steps to build this report 5-79 - 5-81

- Control break (feature)
  - how to build 4-13, 4-22
  - reports that use it 3-2
- Control break report
  - See Break report
- Convert reports from V1.0 to V1.1
  - See MOVEREF
  - See MOVEREP 2-31
- Creating groups with no printable fields
  - brief description of 3-30
  - concepts of 5-7
  - distinguishing features 5-6
  - explanation of 5-6
  - organization of 5-7
  - sample report output 3-30
  - Steps to build 5-7
- Crosstab print direction (feature)
  - reports that use it 3-2
- Crosstab reports 2-17

## **D**

- DataType of PRT (feature)
  - how to build 5-17
- Default report
  - See Tabular report
- Display format
  - example 1-17
- Display formats, altering (feature)
  - how to build 5-39, 5-46
  - reports that have altered display formats 3-2
- DUMPREF command 2-30

## **E**

- Editing field labels 1-13
- Editing text objects 2-25
  - status indicator 2-25
- Error help 1-3
- Execute 1-8
- Executing a report 1-8

## **F**

- Features
  - list of all features in this book 3-2
- Field labels (deleting)
  - how to build 4-26
- Field labels (feature)
  - how to modify 4-15, 4-22
- Field labels, deleting (feature)
  - reports that have deleted field labels 3-2
- Field labels, modifying (feature)
  - how to build 5-24, 5-46
  - reports that contain modified labels 3-2
- Field objects 2-10
  - computed fields 2-11
  - formatting settings 2-11
  - positioning 2-10
  - positioning settings 2-10
  - structural settings 2-10
- Field screens
  - using 1-12, 1-23, 1-31
- Field, computed (report)
  - See Computed field report
- Fields, computed (feature)
  - See Computed field
- Fields, skipping (feature)
  - reports that Skip fields 3-3
- Fields, summary
  - See Summary fields
- Fields, Variable (feature)
  - reports that use them 3-3
- Fields, Wrap (feature)
  - reports that use them 3-3
- First function (feature)
  - how to build 5-26
  - reports that use it 3-2
- Footers, creating (feature)
  - reports that contain footers 3-2
- Footing report
  - group footers 3-27

- Form letter report
  - brief description of 3-36
  - concepts of 4-25
  - distinguishing features 4-24
  - explanation of 4-24
  - method one 4-26
  - method two 4-28
  - organization of 4-25
  - sample report output 3-36
  - Steps to build 4-25, 4-27 - 4-29
- Forward reference 2-21
- Function, First (feature)
  - reports that use it 3-2
- Function, Last (feature)
  - reports that use it 3-2

## **G**

- GENREP command 2-31
- Group footer 2-23
- Group Footing report
  - brief description of 3-27
  - concepts of 4-46 - 4-47
  - distinguishing features 4-46
  - explanation of 4-46
  - organization of 4-47
  - sample report output 3-27
  - steps to build 4-47 - 4-48
- Group header 2-23
- Group objects 2-13
  - data paths 2-17
  - definition of 1-10
  - group settings 2-14
  - group texts 2-23
  - grouping data 1-14
  - matrix reports 2-17
  - multiple groups 2-14
  - pagination 2-16
  - positioning 2-14
  - records 2-14
  - relationship to queries 2-13
- Group text objects 2-23

## **H**

- Highlighting text 1-34

## **I**

- Integrating SQL\*ReportWriter with SQL\*Forms 6-2
- Intermixing fields and groups
  - brief description of 3-15
  - concepts of 5-3
  - distinguishing features 5-3
  - explanation of 5-3
  - organization of 5-4
  - sample report output 3-15
  - Steps to build 5-4
- Introductory mailing label report
  - brief description of 3-34
  - Concepts of 4-16 - 4-17
  - distinguishing features 4-16
  - explanation of 4-16
  - organization of 4-17
  - sample report output 3-34
  - steps to build 4-17 - 4-19
- Introductory mailing labels (feature)
  - reports that use them 3-2
- Invoice report
  - brief description of 3-40
  - concepts of 5-53
  - distinguishing features 5-52
  - explanation of 5-52
  - organization of 5-54
  - sample report output 3-40
  - steps to build 5-54, 5-56

## **L**

- label, modifying (feature)
  - reports that contain modified labels 3-2
- Labels, deleting (feature)
  - reports that contain deleted labels 3-2
- Labels, for mailing
  - See Advanced mailing labels
  - See Introductory mailing labels
- Last function (feature)
  - how to build 5-26
  - reports that use it 3-2
- Loading advanced reports
  - Steps to do this 5-2
- Loading introductory reports
  - Steps to do this 4-2
- LOADREP command 2-30
- Logon procedure 1-5
  - See also SQLREP command

## **M**

- Mailing label report
  - See Advanced Mailing Label Report
  - See Introductory mailing label report
- Margin (feature)
  - reports that use it 3-3
- Master/Detail relationships (feature)
  - pins vs. multiple queries 440
  - reports that use them 3-3
- Master/Detail report
  - brief description of 3-19
  - concepts of 4-38
  - distinguishing features 4-38
  - explanation of 4-38
  - organization of 4-39
  - sample report output 3-19
  - Steps to build 4-39 - 4-40

- Master/Detail/Summary report
  - brief description of 3-21
  - concepts of 4-41 - 4-42
  - distinguishing features 4-41
  - explanation of 4-41
  - organization of 4-42
  - sample report output 3-21
  - steps to build 4-43 - 4-45
- Master/Master report
  - brief description of 3-18
  - concepts of 4-35
  - distinguishing features 4-35
  - explanation of 4-35
  - organization of 4-35
  - sample report output 3-18
  - Steps to build 4-36 - 4-37
- Matrix break report
  - brief description of 3-23
  - concepts of 5-66
  - distinguishing features 5-66
  - explanation of 5-66
  - organization of 5-67
  - sample report output 3-23
  - steps to build this report 5-67, 5-69 - 5-70
- Matrix Group, using (feature)
  - reports that use Matrix Groups 3-2
- Matrix ranking report
  - explanation of 4-34
  - Steps to build 4-34
- Matrix report
  - brief description of 3-22
  - concepts of 4-30
  - distinguishing features 4-30
  - explanation of 4-30
  - organization of 4-31
  - sample report output 3-22
  - steps to build 4-32 - 4-33
- Matrix reports 2-7, 2-17

- Matrix with zeros for null values
  - brief description of 3-24
  - concepts of 5-63
  - distinguishing features 5-63
  - explanation of 5-63
  - organization of 5-64
  - sample report output 3-24
  - steps to build 5-64 - 5-65
- MOVEREP, conversion program 2-31
- Multiple groups 2-14
- Multiple queries 2-5

## **N**

- Non-standard functions
  - See user exits
- Null queries
  - for related groups, brief description of 3-29
  - for related groups, sample output 3-29
  - for unrelated groups, brief description of 3-28
  - for unrelated groups, sample output 3-28
- Null queries (feature)
  - reports that use them 3-3

## **O**

- Outputs, sample
  - list of all outputs in this book 3-4

## **P**

- Page Break of Always (feature)
  - how to build 4-26
  - reports that use it 3-3
- Page footer 2-22
- Page headers (feature) 2-22
  - creating 1-15
  - how to build 4-50

- Page heading report
  - brief description of 3-25
  - concepts of 4-49
  - distinguishing features 4-49
  - explanation of 4-49
  - organization of 4-50
  - See also Placing database values in page headings
    - sample output 3-25
    - Steps to build 4-50
- Pagination 2-16
- Panel 2-16, 2-26
- Parameter objects 2-29
  - created in SQL statement 2-8
- Parent/Child relationships (feature)
  - how to build 4-39, 4-43
  - pins vs. multiple queries 4-40
  - reports that use them 3-3
- Parent/Child report
  - See Master/Detail report
- Placing database values in page headings
  - brief description of 3-26
  - concepts of 5-25
  - distinguishing features 5-25
  - explanation of 5-25
  - organization of 5-26
  - sample report output 3-26
  - Steps to build 5-26 - 5-27
- Positioning fields 2-10
- positioning groups 2-14
- Positioning text 2-27
- Print Direction of Down/Across (feature)
  - how to build 4-18
- Print Group 2-19
- Print Group of a group (feature)
  - reports that use it 3-3
- Print Group of Page (feature)
  - reports that use it 3-3
- Print Group of Report (feature)
  - reports that use it 3-3
- PRINTDEF, printer definition program 2-31

- Printing Reports on Preprinted Forms
  - brief description of 3-39
  - concepts of 5-30 - 5-31
  - distinguishing features 5-30
  - explanation of 5-30
  - organization of 5-32
  - sample report output 3-39
  - steps to build 5-32 -5-34
- PRT datatype (feature)
  - reports that use it 3-3
  - sample output of a report that uses PRT 3-30

## Q

- Queries, null (feature)
  - See reports that use them
- Queries, related
  - Parent/Child relationships 3-3
- Queries, unrelated
  - feature 3-3
- Query objects 2-5
  - multiple queries 2-5
  - parent-child relationships 2-6
- Query screen
  - using 1-7

## R

- Ranking data (feature)
  - reports that perform it 3-3
- Ranking report
  - brief description of 3-12
  - concepts of 5-71
  - distinguishing features 5-71
  - explanation of 5-71
  - organization of 5-72
  - sample report output 3-12
  - steps to build this report 5-72 - 5-73, 5-75
- Record 2-14
- Record Spacing, modifying (feature)
  - how to build 4-18
  - reports that use it 3-3
- Related queries
  - See Parent/Child relationships

- Relative Position of fields (feature)
  - reports that use it 3-3
- Relative Position of groups (feature)
  - how to build 5-21
  - reports that use it 3-3
- Relative Position of Margin (feature)
  - reports that use it 3-3
  - what it is 5-31
- Relative Position of related groups, w/null
  - brief description of 3-29
  - concepts of 5-18 - 5-19
  - distinguishing features 5-18
  - explanation of 5-18
  - organization of 5-20
  - sample report output 3-29
  - steps to build 5-20
- Relative position of unrelated groups, w/null
  - brief description of 3-28
  - concepts of 5-15
  - distinguishing features 5-14
  - explanation of 5-14
  - organization of 5-16
  - sample report output 3-28
  - Steps to build 5-16 - 5-17
- Report elements 2-2
- Report features
  - list of all features in this book 3-2
- Report footer 2-22
- Report header 2-22
- Report name listing 3-4
- Report objects 2-28
  - field 2-10
  - group 2-13
  - parameter 2-29
  - query 2-5
  - report 2-28
  - summary 2-18
  - text 2-22
- Report outputs, sample
  - list of all outputs in this book 3-4
- Reset group 2-19
- Reset Group of a group (feature)
  - reports that use it 3-3
- Reset Group of Report (feature)
  - how to build 4-14
  - reports that use it 3-3

- Run-time parameter form 1-8, 2-30
- Running functions 2-21
- Running totals (feature)
  - reports that use them 3-2
- RUNREP command 2-30

## **S**

- Sample report outputs
  - list of all outputs in this book 3-4
- Skipping fields (feature)
  - reports that Skip fields 3-3
- Spelling out cash amounts on checks
  - brief description of 3-38
  - concepts of 5-60
  - distinguishing features 5-60
  - explanation of 5-60
  - organization of 5-61
  - sample report output 3-38
  - steps to build 5-61 - 5-62
- SQL\*Forms
  - integrating with SQL\*ReportWriter 6-2
- SQLREP command 2-30
- Status indicator 2-25
- Subfooter 2-24
- Subtotal report
  - brief description of 3-7
  - concepts of 4-11 - 4-12
  - distinguishing features 4-11
  - explanation of 4-11
  - organization of 4-13
  - sample report output 3-7
  - steps to build 413 - 4-15
- Summary fields (feature)
  - how to build 4-14, 5-56
  - reports that use them 3-3
- Summary functions 2-20
- Summary objects 2-18
  - creating 1-16
  - forward referencing 2-21
  - summary functions 2-20
- Summary screens
  - using 1-16

- Suppressed column headings report
  - brief description of 3-20
  - concepts 5-22
  - distinguishing features 5-22
  - explanation of 5-22
  - organization of 5-23
  - sample output of 3-20
  - Steps to build 5-23 - 5-24

## **T**

- Tabular report
  - brief description of 3-6
  - concepts 4-3
  - distinguishing features 4-3
  - explanation of 4-3
  - organization of 4-3
  - sample report output 3-6
  - steps to build 4-4
- TERMDEF, terminal definition program 2-31
- Text
  - editing body text 1-27
  - positioning 2-27
- Text objects 2-22
  - editing 2-25
  - group 2-23
  - highlighting 1-34, 2-28
  - justification of 2-27
  - page 2-22
  - pages 2-27
  - panels 2-26
  - positioning 2-27
  - referencing values 2-26
  - report 2-22
- Text screen
  - status 2-25
  - using 1-15, 1-27, 1-33 - 1-34
- Time series calculations report
  - brief description of 3-11
  - concepts of 5-48
  - distinguishing features 5-48
  - explanation of 5-48
  - organization of 5-49
  - sample report output 3-11
  - steps to build 5-49

Title page 2-22  
Totalling columns (feature)  
    See also summary fields  
Totalling rows (feature)  
    See computed fields  
Trailer page 2-22  
Tutorial 1-1

## **U**

Unrelated queries  
    feature 3-3  
user exit (feature)  
    how to use pre-packaged RWECPF 5-80  
    how to use pre-packaged RWEIF 5-73, 5-80.  
User exits (feature)  
    how to use prepackaged RWEIF 5-12, 5-55  
    reports that use them 3-3

## **V**

Variable-length fields (feature)  
    how to build 4-29  
    reports that use them 3-3  
Viewing report output on the screen  
    Browse mode 1-18  
    Window mode 1-18

## **W**

Wide reports 2-16, 2-26  
Window mode 1-18  
Word-wrapped fields (feature)  
    how to build 5-51  
    reports that use them 3-3  
Wrapped break report  
    brief description of 3-14  
    concepts of 5-38  
    distinguishing features 5-38  
    explanation of 5-38  
    organization of 5-39  
    sample report output 3-14  
    steps to build 5-39

Wrapped control break report  
    brief description of 3-14  
    sample report output 3-14  
    See also Break report

## **Z**

Zeros for null values  
    See Matrix report with zeros for null values

# **SQL\*REPORTWRITER® REFERENCE MANUAL**

**VERSION 1.1**

**ORACLE®**

---

The Relational Database Management System

SQL\*ReportWriter Reference Manual Version 1.1

Part No. 641-V1.1 0192

Contributing Author: Karen Denchfield-Masterson

Contributors Steve Breyer, Mylan Bui, Debbie Casanova,  
Dana Hausman, Nimish Mehta, Patty Nghiem,  
Daryl Porter, Jerry Sherman, Dave Simson,  
Larry Stevens, Andre Touma

Copyright © Oracle Corporation 1988,1991

This software/documentation contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this software/documentation is delivered to a U.S. Government Agency of the Department of Defense, then it *is* delivered with Restricted Rights and the following legend is applicable:

#### RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data - General, including Alternate 111 (June 1987).

The information 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.

ORACLE, SQL\*Forms, SQL\*Net, SQL\*Plus, SQL\*ReportWriter, and SQL\*Star are registered trademarks of Oracle Corporation. Oracle\*Mail is a trademark of Oracle Corporation.

Ada is a registered trademark of the U.S. Department of Defense. VMS and DEC are registered trademarks of Digital Equipment Corporation. UNIX is a registered trademark of AT&T Bell Laboratories. HP and LaserJet are registered trademarks of Hewlett-Packard Corporation. PostScript is a trademark of Adobe Systems Inc.



# PREFACE

## **Purpose**

**T**his SQL\*ReportWriter Reference Manual contains a detailed description of the components of SQL\*ReportWriter. This manual does not contain a tutorial or provide conceptual material for users learning how to use SQL\*ReportWriter. For this information, refer to Building Reports with SQL\*ReportWriter.

## **Audience**

The information in this manual is intended primarily for application developers and for readers who want to create reports for their own use. Readers should have a working knowledge of SQL and ORACLE database concepts.

## **How this Manual Is Organized**

The SQL\*ReportWriter Reference Manual is organized into the following Parts:

### **Part I: Using SQL\*ReportWriter**

This section provides information about SQL\*ReportWriters run-time interface, user interface, and on-line help system.

### **Part II: Reference**

This section is organized around each of the SQL\*ReportWriter screens and provides reference information about each attribute on each screen.

## Version 1.0 to 1.1 Changes

For users that are familiar with SQL\*ReportWriter, Appendix F contains a list of new features and their index entries. This will assist those users in learning the new functionality of Version 1.1.

## Related Publications

Along with this manual, you may want to refer to other documents published by Oracle Corporation

- Introduction to SQL\*ReportWriter, Part No. 19678-0688
- Budding Reports with SQL\*ReportWriter, Part No. 5412-V1.1
- SQL\*ReportWriter Installation Guide or System Release Bulletin. This document is different for each hardware/software platform. Ask your sales representative for the appropriate part number.
- SQL Language Reference Manual, Part No. 778-V5.1
- SQL Language Reference Manual, Part No. 778-V6.0
- ORACLE Error Messages and Codes, Part No. 3605-V6.0

## Notational Conventions

The following notational conventions are used in this manual:

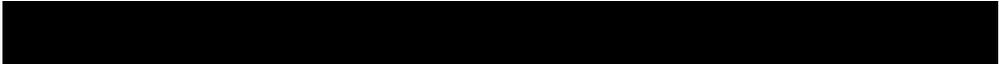
|             |                                                                                                                                                                                                            |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Font Change | Enter text exactly as shown.                                                                                                                                                                               |
| [ ]         | Brackets indicate that the enclosed item is optional. Do not enter the brackets.                                                                                                                           |
| { }         | Curly braces indicate that one, and only one, of the enclosed items must be entered. If the braces are surrounded by brackets, then all of the enclosed items are optional. Do not enter the curly braces. |
|             | A vertical bar is used to separate options within brackets and braces. You must enter one, and only one, of the options separated by the vertical bar. Do not enter the vertical bars.                     |
| —           | The use of underline indicates that the text that is underlined is the default.                                                                                                                            |
| Italics     | The use of italics indicates values or options.                                                                                                                                                            |
| UPPERCASE   | Uppercase words within text indicate command names, keywords, or table names.                                                                                                                              |

**Your Comments  
Are Welcome**

We value and appreciate your comments as an ORACLE user. As we write, revise, and evaluate our work, your opinions are the most important input we receive. At the back of this manual is a Reader's comment Form; we encourage you to use this form to tell us both what you like and dislike about this (or other) Oracle manuals. If the form is gone, or you would like to contact us, please use the following address and phone number:

SQL\*ReportWriter Documentation Manager  
Oracle Corporation  
500 Oracle Parkway  
Redwood Shores, California 94065  
(415) 506-7000





## CONTENTS

### PART I

### USING SQL\*REPORTWRITER

#### CHAPTER 1

#### USER INTERFACE

1-1

SQLREP

1-2

Screen Layout

1-7

SQL\*ReportWriter Screens

1-9

Help System

1-11

Managing Objects

1-13

Editing and Navigation Commands

1-15

#### CHAPTER 2

#### RUN-TIME INTERFACE

2-1

Converting Report Definitions

2-2

RUNREP

2-3

DUMPREP

2-10

GENREP

2-11

LOADREP

2-12

PRINTDEF

2-12

TERMDEF

2-13

MOVEREP

2-13

|                  |                        |            |
|------------------|------------------------|------------|
| <b>PART II</b>   | <b>REFERENCE</b>       |            |
| <b>CHAPTER 3</b> | <b>ACTION MENU</b>     | <b>3-1</b> |
|                  | Action Options         | 3-2        |
| <b>CHAPTER 4</b> | <b>QUERY SCREEN</b>    | <b>4-1</b> |
|                  | Query Management       | 4-2        |
|                  | Default Objects        | 4-3        |
| <b>CHAPTER 5</b> | <b>GROUP SCREENS</b>   | <b>5-1</b> |
|                  | Group Management       | 5-2        |
|                  | Group Screen One       | 5-4        |
|                  | Group Screen Two       | 5-7        |
|                  | Group Screen Three     | 5-10       |
| <b>CHAPTER 6</b> | <b>FIELD SCREENS</b>   | <b>6-1</b> |
|                  | Field Management       | 6-2        |
|                  | Field Screen One       | 6-3        |
|                  | Field Screen Two       | 6-9        |
|                  | Field Screen Three     | 6-15       |
| <b>CHAPTER 7</b> | <b>SUMMARY SCREENS</b> | <b>7-1</b> |
|                  | Summary Management     | 7-2        |
|                  | Summary Screen One     | 7-3        |
|                  | Summary Screen Two     | 7-7        |
| <b>CHAPTER 8</b> | <b>TEXT SCREEN</b>     | <b>8-1</b> |
|                  | Text Objects           | 8-2        |
|                  | Text Management        | 8-3        |
|                  | The Text Screen        | 8-7        |
| <b>CHAPTER 9</b> | <b>REPORT SCREEN</b>   | <b>9-1</b> |
|                  | The Report Screen      | 9-2        |

|                   |                                                                            |             |
|-------------------|----------------------------------------------------------------------------|-------------|
| <b>CHAPTER 10</b> | <b>PARAMETER SCREEN</b>                                                    | <b>10-1</b> |
|                   | Parameter Management                                                       | 10-2        |
|                   | Parameter Screen One                                                       | 10-6        |
|                   | Parameter Screen Two                                                       | 10-10       |
| <b>APPENDIX A</b> | <b>KEY COMMANDS</b>                                                        | <b>A-1</b>  |
|                   | Text Editing Keys                                                          | A-2         |
|                   | Navigation Keys                                                            | A-4         |
|                   | Global Keys                                                                | A-6         |
| <b>APPENDIX B</b> | <b>ADMINISTRATION</b>                                                      | <b>B-1</b>  |
|                   | Upgrading from Previous Versions                                           | B-2         |
|                   | Installing a Userid                                                        | B-3         |
|                   | Dropping Reports                                                           | B-4         |
|                   | Preparing for Sending Reports via Oracle*Mail                              | B-4         |
|                   | Establishing Security and Restrictions                                     | B-4         |
|                   | Memory and Disk Space Requirements                                         | B-5         |
|                   | Extra Disk Space                                                           | B-6         |
|                   | Database Storage Requirements                                              | B-6         |
|                   | File-Searching Method                                                      | B-6         |
|                   | Files Opened During Run-time                                               | B-7         |
|                   | Documenting Your Reports                                                   | B-8         |
| <b>APPENDIX C</b> | <b>SQL*REPORTWRITER TABLES</b>                                             | <b>C-1</b>  |
|                   | Tables Used for both System-owned and User-owned SQL*ReportWriter Tables   | C-2         |
|                   | Additional Objects When System-owned SQL*ReportWriter Tables Are Installed | C-8         |
|                   | Additional Objects When User-owned SQL*ReportWriter Tables Preinstalled    | C-9         |
| <b>APPENDIX D</b> | <b>THE SQL*REPORTWRITER CALL INTERFACE</b>                                 | <b>D-1</b>  |

|                   |                                                 |            |
|-------------------|-------------------------------------------------|------------|
| <b>APPENDIX E</b> | <b>USER EXITS</b>                               | <b>E-1</b> |
|                   | What a User Exit Is                             | E-2        |
|                   | Types of User Exits                             | E-2        |
|                   | Steps Needed for User Exits in SQL*ReportWriter | E-4        |
|                   | Building a User Exit                            | E-4        |
|                   | Linking a User Exit                             | E-9        |
|                   | Passing Arguments to a User Exit                | E-10       |
|                   | Creating an Error Message in a User Exit        | E - n      |
|                   | Returning Values from a User Exit               | E-11       |
|                   | Conditionally Highlighting Fields               | E-12       |
|                   | Restrictions on Writing User Exits              | E-13       |
|                   | User Exits Pre-packaged with SQL*ReportWriter   | E-14       |
| <b>APPENDIX F</b> | <b>VERSION 1.0 TO 1.1 CHANGES</b>               | <b>F-1</b> |
| <b>APPENDIX G</b> | <b>CREATING TERMINAL DEFINITIONS</b>            | <b>G-1</b> |
|                   | Terminal Definition                             | G-2        |
|                   | Tables and Lists                                | G-12       |
| <b>APPENDIX H</b> | <b>PRINTER DRIVERS</b>                          | <b>H-1</b> |
|                   | Printer Driver Definitions                      | H-2        |
|                   | Default Drivers                                 | H-2        |
|                   | Creating/Modifying the Printer Definition File  | H-3        |
|                   | Running the PRINTDEF Command                    | H-5        |
| <b>APPENDIX I</b> | <b>ERROR MESSAGES LISTING</b>                   | <b>I-1</b> |
|                   | Start-up Error Messages                         | I-2        |
|                   | Utility Error Messages                          | I-3        |
| <b>GLOSSARY</b>   |                                                 |            |
| <b>INDEX</b>      |                                                 |            |

PART

**I**

**USING  
SQL\*REPORTWRITER**



# 1

## USER INTERFACE

**I**nformation is entered into SQL\*ReportWriter through its fill-in-the-form interface. This interface allows you to navigate quickly and easily throughout the report definition screens. This chapter discusses the following topics

- SQLREP command
- user interface components
- SQL\*ReportWriter screens
- menus, dialog boxes, and List of values
- navigation among screens
- object management
- on-line help
- editing functions.

## SQLREP

The SQLREP coremand invokes SQL\*ReportWriter for defining and maintaining reports. To interactively view your command line options, type sqlrep ?. Note that the exact syntax of this command is operating system specific. For example, in the UNIX C shell, type Sqrep " ? ".

syntax sqlrep [ [keyword=] value ] . . .

where the following are valid keyword=value arguments

```
[[USERID=]userid]
[[PARAM14FORM=] {yes| no}]
[[CMDFILE=] f i l e name]
[[TERM=] term f i l e]
[[ARRAYSIZE=] n]
[[DESTYPE=] {screen 1 file |printer| mail}]
[[DESNAME=] desname]
[[DESFORMAT=] desformat]
[[COPIES=] n]
[[CURRENCY=] currency_symbol]
[[THOUSANDS=] thousands_symbol]
[[DECIMAL=] decimal_symbol]
[[LOGFILE=]logfile]
[[BUFFERS=]n]
[[LANGUAGE=]language_name]
[<param>=]value]
```

**Note:** You may omit keywords, but then you must enter the arguments in the order in which they appear above.

The following are examples of command line sequences:

```
sqlrepuserid=scott/tiger paramform=no destype=mail desname=rmiller
sqlrep scott/tiger no destype=mail desname=rmiller
```

USERID Allows you to specify your ORACLE user name and password with an optional SQL\*Net address. The logon definition must be in one of the following forms

```
username
username/password
username@node
username/password@node
```

If you omit your password, a logon form is provided. To invoke a default logon procedure (using a username prefixed with OPS\$), do one of the following:

- type `sqlrep /` on the command line
- enter a slash (/) in the logon form as your username
- omit your username/password on the command line and press the Return key twice when the logon form appears.

If any of these procedures is unsuccessful, you are prompted for your username and password. After three unsuccessful attempts, the SQLREP process will automatically terminate.

**PARAMFORM** Allows you to display the Run-time Parameter Form. when you execute a report. The default value is YES, meaning to display the form.

**CMDFILE** Allows you to specify a file which contains a set of arguments for SQLREP. This allows you to specify many arguments, or a standard set of arguments, without typing them in each time you use SQLREP. The CMDFILE may contain any argument except another CMDFILE. The format requirements of the CMDFILE are the same for those of the command line.

If you specify arguments both in a CMDFILE and on the command line, SQL\*ReportWriter uses the arguments it first encounters on the command line. For example, if a CMDFILE called MY.CMD specifies PARAM1=5000, and the command syntax is

```
runrep myreport scott/tiger param1=1000 cmdfile=my. cmd
```

SQL\*ReportWriter will use PARAM1=1000 when running the report.

**TERM** Allows you to specify a file that describes the terminal from which you are using SQLREP. In turn, this file determines which keys correspond to which functions. The default is installation dependent.

**ARRAYSIZE** Allows you to specify the size of the array in kilobytes for use with the ORACLE array interface. The default arraysize is 10.

**DESTYPE** Allows you to specify the type of output device that will receive the report output. The default is Screen. Valid values are:

- Screen routes the output to the browser for interactive viewing
- File routes the output to the file named in DESNAME (the next keyword discussed)

- Printer routes the output to the printer named in DESNAME (the next keyword discussed)
- Mail causes output to be sent to Oracle\*Mail users. If your system does not have Oracle\*Mail installed, or is not an Oracle\*Mail node, an error will be returned. The report is sent as an attached file, not directly as a mail message.

**Note:** Sysout is not a valid option for DESTYPE while using SQLREP. It can only be used with RUNREP when BATCH=YES. To view your report, use Screen.

**DESNAME** Allows you to specify the file, printer, or Oracle\*Mail userid (or distribution list) to which the report output will be sent. To send the report output to an Oracle\*Mail user, specify the userid as you do in Oracle\*Mail. To specify multiple user names to send to, use the syntax: (<name>, <name>, . . .<name>). Note: This keyword is ignored if DESTYPE is Screen.

**DESFORMAT** Allows you to specify the characteristics of the printer named in DESNAME. This keyword is ignored if DESTYPE is Screen. Examples of valid values for this keyword are hpl, hplwide, dec, decwide, decland, dec180, dflt, wide, etc. Ask your System Administrator for a list of valid destination formats.

**COPIES** Allows you to specify the number of copies of the report to print. The default is 1. **Note:** This keyword is ignored for any DESTYPE other than Printer.

**CURRENCY** allows you to specify the currency character to be used in number formats. The default for ORACLE Version 6.0 is taken from the database, using the ORACLE National Language Support facilities. The default for other versions of ORACLE is '\$'. A CURRENCY value entered on the command line will override any LANGUAGE value entered. A CURRENCY value entered on the Parameter Settings screen will override any CURRENCY value entered on the command line.

**THOUSANDS** allows you to specify the thousands indicator to be used in number formats. The default for ORACLE Version 6 is taken from the database, using the ORACLE National Language Support facilities. The default for other versions of ORACLE is ','. A THOUSANDS value entered on the command line will override any LANGUAGE value entered. A THOUSANDS value entered on the Parameter Settings screen will override any THOUSANDS value entered on the command line.

**DECIMAL** allows you to specify the currency character to be used in number formats. The default for ORACLE Version 6 is taken from the database, using the ORACLE National Language Support facilities. The default for other versions of ORACLE is '.'. A DECIMAL value entered on the command line will override any LANGUAGE value entered. A DECIMAL value entered on the Parameter Settings screen will override any DECIMAL value entered on the command line.

**LOGFILE** Allows you to specify which file should receive the output from the interactive print commands [Print Page], [Print Screen], and [Print Report]. If the file already exists, output will be appended to it. The default logfile is dfltrep.log.

**BUFFERS** Allows you to specify the maximum number of virtual memory buffers that will be open at any one time for the SQLREP and RUNREP executable. The size of a single buffer is 512 bytes. As an example, the default for VAX/VMS and Sun UNIX is 50 buffers. The default for all systems is optimal for 'normal' reports (up to a few hundred pages). This number can be increased for larger reports, although increasing it too much might cause degradation of system performance in a multi-user environment.

The limit on the number of buffers is system-dependent. On VMS and UNIX, for example there is no hard limit, but on PC/AT's the limit is 128 (i.e., 64K bytes).

**LANGUAGE** Allows you to specify the language system with which SQL\*ReportWriter should display fields of Data Type of NUM, or DATE. The default is U.S. (e.g., a number might appear as \$555,232.87). Any Default Value for CURRENCY, DECIMAL, or THOUSANDS on the Parameter Screen, Run-time Parameter Form, or command line will override the value entered for the LANGUAGE keyword. Note: This keyword is only meaningful with ORACLE Version 6.0.

**<PARAM>** Allows you to name a parameter created as part of the report definition. The value you enter is assigned to the parameter. If you do not supply a value, the default value, specified on the Parameter Screen, is used. You may also override that default value on the Run-time Parameter Form. See the "Run-time Parameter Form" in Chapter 2.

### **Command Line Syntax Rules**

1. Command line arguments are of the form keyword=value, with no spaces before or after the equal sign.
2. Arguments are separated by one or more spaces, not commas.

3. Values of arguments maybe in single or double quotes. The effect of single or double quotes is operating-system specific.
4. All arguments (i.e., PARAMFORM=YES) are optional.
5. The "keyword=" part of each argument is optional as long as the arguments are entered in the order shown in the syntax. If you skip one of the arguments, you must use keywords. For example, if you had a command line that contained four keywords and you wanted to specify the first, second, and fourth arguments, you could specify the first two without the keywords, but would have to specify the fourth argument with a keyword (otherwise it would think that the last argument was really the third argument). Note Once a keyword is used in a command line, the remainder of the command line must contain keywords. If arguments are entered in any other order, keywords must be supplied.
6. The "keyword=" part of all arguments is not case sensitive. The "value" portion of the CMDFILE, TERM, DESNAME, DESFORMAT, and LOGFILE arguments maybe case sensitive, depending on the operating system. The "value" portion of all other arguments is not case sensitive.
7. More than one user with the same ORACLE username can run SQL\*ReportWriter simultaneously if the users are using system-owned SQL\*ReportWriter tables. If user-owned tables are being used, only one person can run SQLREP under a single ORACLE username at a time. If you attempt to use SQLREP after someone else already has begun a session with the same ORACLE username, an error message is displayed and your session is terminated. This restriction does not affect the use of SQLREP with multiple ORACLE usernames, or the use of RUNREP.
8. Full pathnames are supported for all file references.
9. The LANGUAGE value is overridden by CURRENCY, THOUSANDS, and DECIMAL values.

### **Open Files During Run-time**

For a list of the files that maybe opened at some time during run-time, see "Administration" in Appendix B for details.

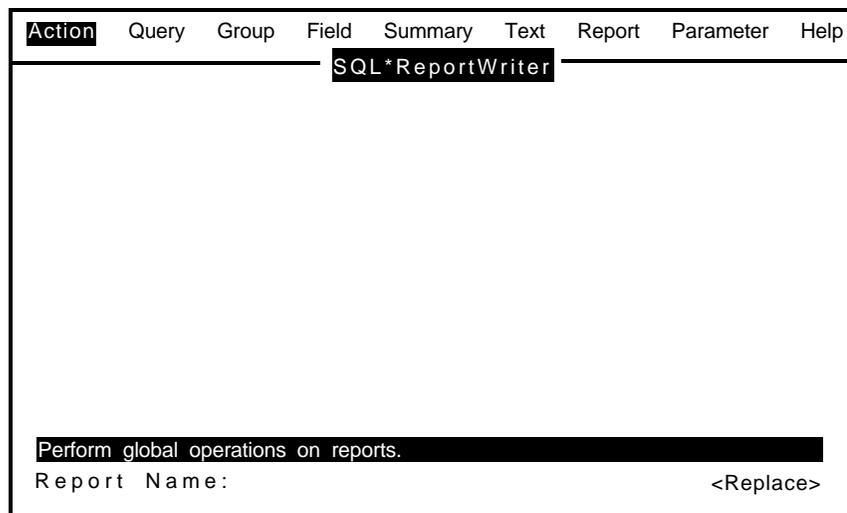
## Screen Layout

This section describes the basic parts of the user interface, that is, the components that are common to all SQL\*ReportWriter screens.

## Main Menu

The Main Menu appears when SQL\*ReportWriter is invoked, when [Accept] is pressed from any setting screen or menu, or when [Menu] is pressed. See Figure 1-1.

**FIGURE 1-1**  
**The Main Menu**



## Menu Line

The first line on the screen is the Menu Line. The Main Menu appears here and contains these choices

**Action**                      Action is a pull-down menu with a number of options allowing you to perform various file and report-level operations.

|                  |                                                                                                                                                                                                                                                                                    |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Query</b>     | The Query Screen is used to enter SELECT statements and create relationships between multiple queries.                                                                                                                                                                             |
| <b>Group</b>     | The Group Screens are used to specify the positioning, hierarchy, and appearance of groups.                                                                                                                                                                                        |
| <b>Field</b>     | The Field Screens are used to assign fields to groups for positioning and layout, to determine specific attributes for individual fields, and to create computed fields.                                                                                                           |
| <b>Summary</b>   | The Summary Screens are used to specify periodic and running summaries for field                                                                                                                                                                                                   |
| <b>Text</b>      | The Text Screen is used to specify text for page, report, and group objects, including headers, footers, title pages and trailer pages.                                                                                                                                            |
| <b>Report</b>    | The Report Screen is used to set page dimensions, the Run-time Parameter Form default title, hint, and status, and Access privileges (when using system-owned SQL*ReportWriter tables). You can also enter comments about the report, and view your report history on this screen. |
| <b>Parameter</b> | The Parameter Screen is used to specify values for parameters embedded in reports. It is also used to modify default report output information (i.e., if the output should be sent to the screen or a printer).                                                                    |
| <b>Help</b>      | The Help choice displays a screen which contains information about using the help system; from that screen you can access the components of the help system.                                                                                                                       |

**Title Line** The second line on the screen is the Title Line. This line displays the title of the current screen. In the help system, the topic of the help screen is displayed in this line.

**Work Area** The work area begins on the third line. In the Main Menu this area is empty; a spreadsheet or form appears herein the other SQL\*ReportWriter screens.

**Hint Line** The first line below the work area is the Hint Line. This line displays a brief hint that prompts for the appropriate action. The hint changes as the cursor moves from attribute to attribute.

The last line on the screen is the Status Line. The status line displays the following information

- the name of the current report when one is open
- whether or not the current field has a List of values
- (see List of values later in this chapter for more information)
- the status of the insert/replace toggle for editing purposes
- the status of the Query mode toggle and the highlight indicator (see the Text Screen chapter for more information).

## **Error Area**

In the event of an error, the error area will cover the hint and status lines. In some cases, the error message is scrollable, and you can press [Previous Line], [Next Line], [Left], and [Right] to reveal more information about your error. Pressing any other key, when an error is displayed, will remove the error message from the screen.

---

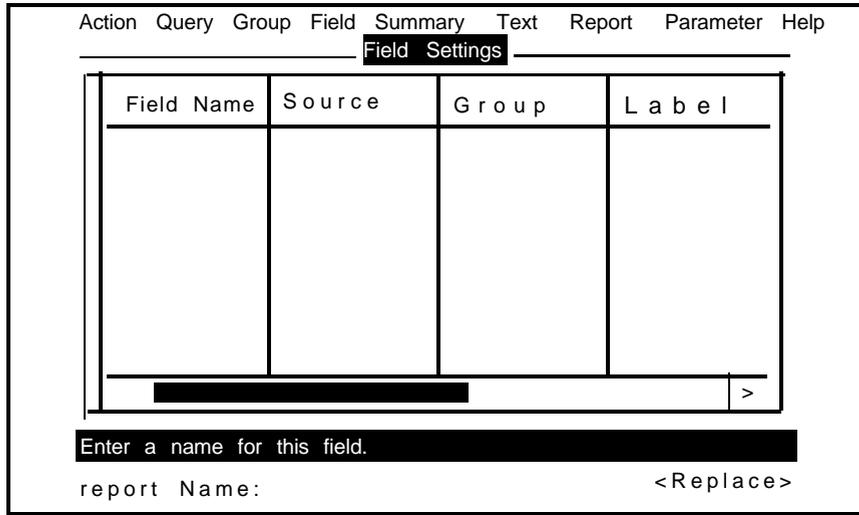
## **SQL\*ReportWriter Screens**

Reports are defined by filling in simple forms called settings screens. There are seven kinds of screens

- Query
- Group
- Field
- summary
- Text
- Report
- Parameter.

Figure 1-2 is an example of a SQL\*ReportWriter screen. Screens are composed of settings and records. A setting is an area into which text is entered. A record is a series of settings.

**FIGURE 1-2**  
**A SQL\*ReportWriter Screen**



**Option Fields**

Some field settings accept only “X” or a blank as valid values. These fields are referred to as option fields. Enter “X” to enable the option; a blank cancels the option.

**Menus**

A menu is a list of available choices that allows you to determine what action to take next or what element to go to next.

**List of Values**

A List of values is a list of all valid values for the current field. The status line displays <List> if a List of values is available for that field.

Display the List of values for a field by placing the cursor in that field and pressing [List]. Scroll through the list with [Next Choice] and [Previous Choice] or [Scroll Up] and [scroll Down], and pick one of the values by placing the cursor on the desired value and pressing [Select].

On the Query Screen, there are two Lists of values that are available in the SELECT Statement area. The first List of values displays all tables from which you may choose. Invoke this list by pressing [List] once. You may view the list, move your cursor to a table name and press [Select] to insert it in your SELECT Statement, or you may use the table list to invoke a second list: a list of all columns for a table in which you are interested. To exit the tables list, press [Undo].

To invoke the list of columns for a particular table, move the cursor to a table name and press [List]. You may simply view the list, or you may move your cursor to a column you want to insert into your SELECT Statement and press [Select]. The column will be inserted where your cursor was in your SELECT Statement. You may select any number of columns from any number of tables from the lists. To exit the columns list, press [Undo].

The SELECT Statement area on the Query Screen is the only place where a List can invoke another List.

### **Dialog Box**

A dialog box is a window that appears in the center of the screen to request input from the user.

### **Alert Box**

An alert box is a window that appears in the center of the screen to offer a list of options; you must choose one to complete the command. Use [Next Choice] and [Previous Choice] to highlight the desired option and then press [Select], or simply press the first letter of the choice.

### **Elevators**

An elevator is an indicator of the number of screens. There are two elevators on the SQL\*ReportWriter screens: a horizontal elevator and a vertical elevator. The vertical elevator, identified by its up arrow and down arrow, is located on the left-hand side of the screen. The horizontal elevator, identified by its bold, solid bar, is located at the bottom portion of the screen. If there is more than one screen, the bar will be shorter than the entire length of the screen in proportion to the number of screens available (i.e., if there are four screens, the bar will be one-fourth the screen length). If you move to the next screen, the bar will move one proportion over (i.e., if there are four screens and you move to the second, the bar will move one-fourth of a position to the right).



### **Help System**

To get help for any part of SQL\*ReportWriter, place the cursor on any field or menu choice and press [Help]. A full screen of information is displayed appropriate to the current setting or choice. Occasionally, the information is displayed on more than one screen: press [Scroll Up] or [Scroll Down] to move from one screen to another within a topic. The help system is organized by chapters, and is reflected in the on-line Table of Contents.

## Help Menu

The help system has its own menu, shown on the menu line. The menu allows you to navigate to various parts of the help system, and contains the following options:

|           |                                                      |
|-----------|------------------------------------------------------|
| Previous  | Go to the previous help topic.                       |
| Next      | Go to the next help topic.                           |
| MainTopic | Go to the chapter screen for the current topic.      |
| Index     | Go to the alphabetical index of topics.              |
| Contents  | Go to the topically organized table of contents.     |
| Examples  | Go to an example that illustrates the current topic. |
| Quit      | Return to the definition screen.                     |

## Keywords

Embedded in the text of each help screen are keywords that identify related topics. These words are highlighted. To view a screen about a related topic, place the cursor on a keyword and press [Select]. Note that first letter picks are available only for menu choices, not keywords.

## Other Help Features

The following are additional help system features which allow you even more flexibility. We suggest you read the "Help About Help" screen (see below) before you use the system for the first time.

## Navigation in the Help System

Use [Scroll Up] and [Scroll Down] to navigate through pages of a topic. [Scroll Top] and [Scroll Bottom] take you to the first and last pages, respectively.

**Backtrack and Bookmark** Press [Undo] to backtrack to the prior help screen. Press [Bookmark] to place a mark on the current help topic and *return* to the report definition. Press [Bookmark] again to return to the marked topic. Do not confuse Bookmark with the Mark command, which is used for cutting and pasting text.

## Key Help

At any point in SQL\*ReportWriter you can get a listing of all valid keys in the current context by pressing [Key Help].

## Help About Help

To see a help screen that gives an overview of the help system and how to use it, select Help from the Main Menu, or, press [Help] while you are viewing any help screen.

## **Error Help**

Help screens are available any time you receive an error message to explain why you received the error and how to recover from it. To view the help screen for your error, press [Help] while the error message is displayed on the hint line.

If you have made a SQL error, an asterisk will appear in the error message at the point at which the error occurred. SQL error messages are several lines long, so you may need to press [Next Line] several times to see the exact line in your SQL Statement where the error occurred.

If you have made an error and wish to discard your changes to the current screen, press [Undo]. On some keyboards you must press a key twice to execute the Undo function.

Note: If you receive an error message in the form "Error -XXXX reported from ORACLE," this is equivalent to error ORA-XXXX in the ORACLE Error Messages and Codes manual.

## **Examples**

Some help screens have an associated screen which displays an example of the feature in use. The Example choice on the Help Menu is displayed in bold if it is active. To select the example choice, place the cursor on the word Example and press [Select].

## **Managing Objects**

Most objects (field, summary, query, group, text panel, parameter) in a report definition can be inserted, deleted, and renamed. All objects except queries can be moved.

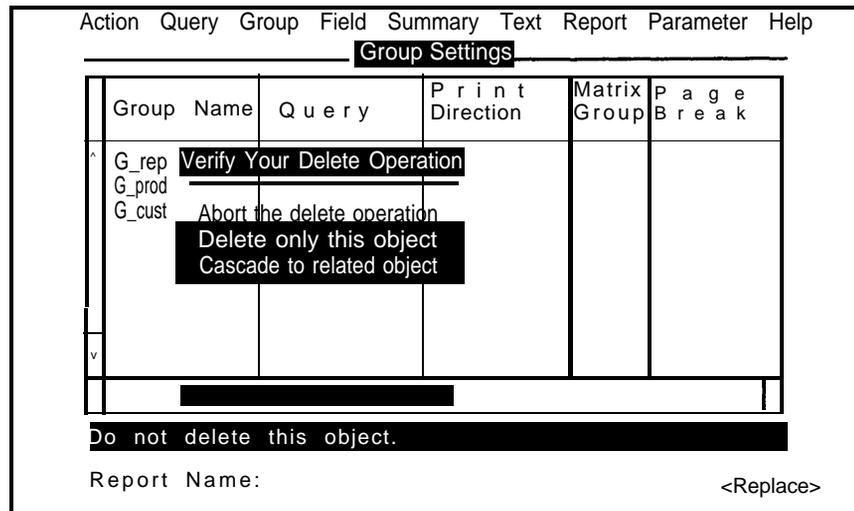
### **Inserting an Object**

Press [Insert Record Below] or [Insert Record Above] and define the new object by filling in the settings. When you insert fields, groups, and summaries, a blank line is opened; on the Query and Text screens, an empty form is displayed. The information on each Query or Text form is one record; when you press [Next Record] or [Previous Record], a new set of information is displayed on the form.

### **Deleting an Object**

Navigate to the object to be deleted and press [Delete Record]. Many objects in a report depend on other objects. For example, the fields in a report depend on the group they belong to. When an object that is referenced by another object is deleted, SQL\*ReportWriter takes note of the fact and displays an alert box which offers the three alternatives shown in Figure 1-3.

**FIGURE 1-3**  
Delete Operations Alert Box



- Abort the delete operation cancels the delete.
- Delete only this object deletes the object (group, field, summary, parameter) by placing it in a delete buffer, but does not modify any references to the object.
- Cascade to related objects deletes the object and all objects which refer to it.

### Moving an Object

Follow the steps below to move an object from one position to another.

1. Place the cursor on the appropriate object and press [Delete Record]. Select the "Delete..." choice from the alert box.
2. Move the cursor to the new position and press [Undelete Record]. The current object moves down one line to make room for the moved object.

### Renaming an Object

Move to the name of the object to be renamed and type in the new name. SQL\*ReportWriter automatically updates all references to that object.

### Undefined Objects

Undefined references occur when an object that is referenced by other objects is deleted using the "Delete..." option of the delete alert box. Reports that contain undefined references cannot be executed. To correct the references, replace 'Uncle fined' with a valid name by selecting the appropriate choice from the List of values.

## **Editing and Navigation Commands**

SQL\*ReportWriter contains a full set of editing commands, including shortcuts for moving a character, word, or line at a time, as well as cutting and pasting, and undeleting (see Appendix A). Commands are provided for the following functions:

- text editing
- navigation
- global operations
- file import and export
- search and replace
- auto-repeat.

### **Copying Text**

To copy text from a report to the buffer, to be read out to a text file or moved to another part of the report, do the following:

1. Place the cursor on the first letter of the section of text you want to copy and press [Mark].
2. Move the cursor one character beyond the last letter of the text you want to copy. The text that will be copied is displayed in reverse-video. Press [Copy]. The specified text is copied into the buffer.

### **Deleting Text**

To delete text from a report, do the following:

1. Place the cursor on the first letter of the section of text you want to delete and press [Mark].
2. Move the cursor one character beyond the last letter of the text you want to delete and press [Cut].

Once you have deleted the text, you may then paste or write the text to a file, or leave it in the delete buffer. The delete buffer will be erased when the current SQL\*ReportWriter session is terminated.

Note: The delete buffer always contains the text deleted by the most recent delete or copy operation.

### **Pasting Text**

To copy text from the buffer into a report, place the cursor where you want the first letter of the text to be inserted and press [Paste]. [Paste] is usually used in conjunction with [Cut] or [Copy], both of which put text into the buffer.

### **Reading Text from a File**

This function is usually used for reading in boilerplate text or long queries. To read from any text file into a report, do the following:

1. Place the cursor where you want the first letter of the text and press [Read File].
2. At the prompt, enter the name of the text file (which may include a path name) and then press the [Accept] key; all the text from that file is read into the report.

### **Writing Text to a File**

To copy text from the report into a text file, do the following

1. Place the cursor on the first letter of the text you want to copy and press [Mark].
2. Move the cursor to the last letter of the text you want to copy. The text that will be copied is displayed in reverse-video. Press [Copy]. You can also press [Cut] if you want to remove the text.
3. Press [write File].
4. At the prompt, enter the name (and pathname if necessary) of the file you want to copy the text into and press [Accept].

# RUN-TIME INTERFACE

**T**his chapter discusses the RUNREP, GENREP, DUMPREP, LOADREP, and PRINTDEF commands, including descriptions of all their command line options. This chapter describes how to:

- run reports directly using the RUNREP command in either batch or interactive mode
- provide values for parameters at run-time (RUNREP)
- view report output interactively (RUNREP)
- generate a runfile (GENREP)
- transfer a report definition into an ASCII file (DUMPREP)
- load an ASCII file containing a report definition into an ORACLE database (LOADREP)
- create a format file for a particular printer (PRINTDEF)
- create a format file for a particular terminal (TERMDEF)
- convert reports from SQL\*ReportWriter Version 1.0 to 1.1 (MOVEREP).

## Converting Report Definitions

The following chart displays which commands are needed to convert report definitions from one format to another.

| <b>From -</b>             | <b>To</b>                 |                           |                     |
|---------------------------|---------------------------|---------------------------|---------------------|
|                           | <b>&lt;report&gt;.rep</b> | <b>&lt;report&gt;.rex</b> | <b>ORACLE RDBMS</b> |
| <b>&lt;report&gt;.rep</b> | ---                       | N/A                       | N/A                 |
| <b>&lt;report&gt;.rex</b> | GENREP                    | ---                       | LOADREP             |
| <b>ORACLE RDBMS</b>       | GENREP<br>SQLREP          | DUMPREP                   | ---                 |

where:

<report>.rep is a binary report definition file needed to run reports via RUNREP.

<report>.rex is an ASCII or EBCDIC report definition file needed to move report definitions across different hardware and software platforms.

ORACLE RDBMS implies a report definition that is stored in an ORACLE database.

For information on upgrading from previous versions of SQL\*ReportWriter and/or ORACLE, please refer to Appendix B.

## RUNREP

The RUNREP command allows you to execute previously defined reports by entering a command from the operating system. RUNREP also allows reports to be executed in batch mode, without user intervention. You must have generated a runfile before you can execute the report. To interactively view your command line options, just type runrep ?. Note that this is operating system specific. For example, in the UNIX csh shell, type run rep "?".

syntax runrep [report=] runfile [ [keyword=] value . . . ]

where the following are valid keyword=value arguments:

```
[[USERID=] userid J
[[PARAMFORM=] {YES | NO} 1
[[CMDFILE=] cmdfile]
[[TERM=] termfile]
[[ARRAYSIZE=] n]
[[DESTTYPE=] {screen | file | printer | sysout | mail}]
[[DESNAME=J desname]
[[DESFORMAT=] desformat]
[[COPIES=] n]
[[cuRREncy=] currency_symbol]
[[THOUSANDS=] thousands_symbol]
[[DECIMAL=] decimal_symbol]
[[READONLY=]{YES | NO}]
[[LOGFILE=] logfile]
[[BUFFERS=]n]
[[LANGUAGE=] language_name]
[[BATCh=] {YES | NO}]
[<param>=value]
```

**Note:** If you omit the key words, you must enter the arguments in the order in which they appear above.

The following are examples of command line sequences

```
runrep userid=scott/tiger report=myreport cmdfile=mycmd.in
runrep myreport scott/tigermymcmd.in readonly=yes
```

REPORT Identifies the report to execute. A runfile must be specified. There is no default.

**USERID** Allows you to specify your ORACLE username and password with an optional SQL\*Net address. The logon definition must be in one of the following forms:

|               |                         |
|---------------|-------------------------|
| username      | username/password       |
| username@node | username/password@ node |

To invoke a default logon procedure (using a username prefixed with OPS\$), do one of the following

- type run rep / on the command line
- enter a slash (/) in the logon form as your username
- omit your username/password on the command line and press the Return key twice when the logon form appears.

If any of these procedures is unsuccessful, you are prompted for your username and password. If you omit your password three times, you are prompted for it.

**PARAMFORM** Allows you to display the run-time parameter form when you execute a report. The default value is YES, meaning to display the form.

**CMDFILE** Allows you to specify a file which contains a set of arguments for RUNREP. This allows you to specify many arguments, or a standard set of arguments, without typing them in each time you use RUNREP. The CMDFILE may contain any argument except another CMDFILE. The format requirements of the CMDFILE are the same for those of the command line. See the command line syntax rules at the end of this section.

If you specify arguments both in a CMDFILE and on the command line, SQL\*ReportWriter uses the arguments it first encounters on the command line. For example, if a CMDFILE called MY.CMD specifies PARAM1=5000, and the command syntax is

```
runrep my report scott/tiger param1=1000 cmdfile=my. cmd
```

SQL\*ReportWriter will use PARAM1=1000 when running the report.

**TERM** Allows you to specify a terminal definition file that describes the terminal from which you are using RUNREP. In turn, this file determines which keys correspond to which SQL\*ReportWriter functions. The default is installation dependent. (See your System Administrator for a compatible definition.)

**ARRAYSIZE** Allows you to specify the size of the array in kilobytes for use with the ORACLE array interface. Generally, the larger the ARRAYSIZE, the faster the report will run. The default arraysize is 10.

**DESTYPE** Allows you to specify the destination device that will receive the report output. The default value is taken from the DESTYPE parameter on the Parameter Screen (the default is *Screen* if there is no value). Valid values are

- Screen writes the output for interactive viewing using the browser
- File causes the output to be routed to the file named in DESNAME
- Printer causes output to be routed to the printer named in DESNAME
- Mail causes output to be sent to Oracle\*Mail users. If your system does not have Oracle\*Mail installed, or is not an Oracle\*Mail node, an error will be returned. The report is sent as an attached file, not directly as a mail message
- Sysout causes output to be routed to the default system output device. This value is valid only when the BATCH=YES argument was used in the command line.

**DESNAME** Allows you to specify the file, printer, or Oracle\*Mail userid (or distribution list) to which the report output will be sent. The default value is taken from the DESNAME parameter on the Parameter Screen. To send the report output to an Oracle\*Mail user, specify the userid as you do in Oracle\*Mail. To specify multiple user names to send to, use the syntax: (<name>, <name>, . . . <name>). Note: This keyword is ignored if DESTYPE is Screen.

**DESFORMAT** Allows you to specify the characteristics of the printer named in DESNAME. This keyword is ignored if DESTYPE is Screen. The default value is taken from the DESFORMAT parameter on the Parameter Screen. Examples of valid values for this keyword are hpl, hplwide, dec, decwide, dft, wide, etc. Ask your System Administrator for a list of valid destination formats.

**COPIES** Allows you to specify the number of copies of the report to print. The default value is taken from the COPIES parameter on the Parameter Screen (the default is 1 if there is no value). Note: This keyword is ignored for any DESTYPE other than Printer.

**CURRENCY** allows you to specify the currency character to be used in number formats. The default for ORACLE Version 6.0 is taken from the database, using the ORACLE National Language Support facilities. The default for other versions of ORACLE is '\$'. A CURRENCY value entered on the command line will override any LANGUAGE value entered. A CURRENCY value entered on the Parameter Settings screen will override any CURRENCY value entered on the command line.

**THOUSANDS** allows you to specify the thousands indicator to be used in number formats. The default for ORACLE Version 6 is taken from the database, using the ORACLE National Language Support facilities. The default for other versions of ORACLE is ','. A THOUSANDS value entered on the command line will override any LANGUAGE value entered. A THOUSANDS value entered on the Parameter Settings screen will override any THOUSANDS value entered on the command line.

**DECIMAL** allows you to specify the currency character to be used in number formats. The default for ORACLE Version 6 is taken from the database, using the ORACLE National Language Support facilities. The default for other versions of ORACLE is '.'. A DECIMAL value entered on the command line will override any LANGUAGE value entered. A DECIMAL value entered on the Parameter Settings screen will override any DECIMAL value entered on the command line.

**READONLY** Allows you to request read consistency across queries in a multiple query report. When accessing data from ORACLE Version 6.0, read consistency is accomplished by a SET TRANSACTION READ ONLY statement (refer to the SQL Language Reference Manual for more information on SET TRANSACTION READ ONLY). When accessing data from ORACLE Version 5.0, the data tables are locked in SHARE MODE. The READONLY keyword is only useful for reports using multiple queries, because ORACLE automatically provides read consistency, without locking for single query reports. The default value for this keyword is NO, i.e. do not provide read consistency for multiple query reports.

**LOGFILE** Allows you to specify which file should receive the output from the interactive print commands: [Print Page], [Print Screen], and [Print Report]. If the file already exists, output will be appended to it. The default logfile is dfltrep.log.

**BUFFERS** Allows you to specify the maximum number of virtual memory buffers that will be open at any one time for the SQLREP and RUNREP executable. The size of a single buffer is 512 bytes. As an example, the default for VAX/VMS and Sun UNIX is 50 buffers. The default for all systems is optimal for 'normal' reports (up to a few hundred pages). This number can be increased for larger reports, although increasing it too much might cause degradation of system performance in a multi-user environment.

The limit on the number of buffers is system-dependent. On VMS and UNIX, for example there is no hard limit, but on PC/AT's the limit is 128 (i.e., 64K bytes).

**LANGUAGE** Allows you to specify the language system with which SQL\*ReportWriter should display fields of Data Type of NUM, or DATE. The default is U.S. (e.g., a number might appear as \$555,232.87). Any value for CURRENCY, DECIMAL, or THOUSANDS on the Parameter Screen, Run-time Parameter Form, or command line will override the value entered for the LANGUAGE keyword. **Note:** This keyword is only meaningful with ORACLE Version 6.0.

**BATCH** Allows you to suppress all terminal input and output, in order to run reports without user intervention. The default value is NO, which allows special terminal input and output. If BATCH is enabled, PARAMFORM=YES is invalid because it is not meaningful to have the Run-time Parameter Form appear in batch mode. **Note:** You must specify BATCH=YES if you invoke RUNREP from a script. If you wish to invoke RUNREP to execute a report in the background without BATCH=YES, you must enter the appropriate operating system command at the operating system prompt. See your Installation and User's Guide for more information.

<**PARAM**> Allows you to name a parameter created as part of the report definition. The value you enter is assigned to the parameter and appears on the optional Run-time Parameter Form. If you do not supply a value, the default value, specified on the Parameter Settings Screen, is used. You may also override that default value on the Run-time Parameter Form. See the "Run-time Parameter Form" later in this chapter.

### Command Line Syntax Rules

1. Command line arguments are of the form keyword=value, with no spaces before or after the equal sign.
2. Arguments are separated by one or more spaces, not commas.
3. Values of arguments maybe in single or double quotes. The effect of single or double quotes is operating-system specific.
4. All arguments (e.g., PARAMFORM=YES) are optional, except runfile.
5. The "keyword=" part of each argument is optional as long as the arguments are entered in the order shown in the syntax. If you skip one of the arguments, you must use keywords. For example, if you had a command line that contained four keywords and you wanted to specify the first, second, and fourth arguments, you could specify the first two without the keywords, but would have to specify the fourth argument with a keyword (otherwise it would think that the last argument was really the third argument). **Note:** Once a keyword is used in a command line, the remainder of the command line must contain keywords. If arguments are entered in any other order, keywords must be supplied.

6. The “keyword=” part of all arguments is not case sensitive. The “value” portion of the CMDFILE, TERM, DESNAME, DESFORMAT, and LOGFILE arguments maybe case sensitive, depending on the operating system The “value” portion of all other arguments is not case sensitive.
7. Full pathnames are supported for all file references.
8. LANGUAGE is overridden by CURRENCY, THOUSANDS, and DECIMAL.

### **Open Files During Run-time**

For a list of the files that maybe opened at some time during run-time, see “Administration” in Appendix B for details.

### **Run-time Parameter Form**

You can pass parameter values to RUNREP from the operating system command line at run-time by using <param>=value arguments. These values can be overridden by the Run-time Parameter Form (see Figure 2-1). The Run-time Parameter Form appears when you execute a report using the Action menu, or specify RUNREP from the command line.

The Run-time Parameter Form allows you to override Print options and parameter values you specified on the Parameter Screen or via <para~>=value arguments. To use the values that appear on the Form, press [Accept] without making changes. To quit without ruining the report, press [Undo] and the report will not execute. To suppress the display of this form, specify PARAMFORM=NO on the command line for either SQLREP or RUNREP.

**FIGURE 2-1**  
**Run-Time Parameter Form**

| Parameter Values         |                          |
|--------------------------|--------------------------|
| Parameter                | Value                    |
| Destination Type         | Screen                   |
| File Name/Spool Device   | Master_Detail_Report.lis |
| Printer Description File | dflt                     |
| Number of copies         | 1                        |

Enter the desired value for each parameter.

Report Name: Master\_Detail\_Report <Replace>

### **Browse Mode**

When the output of the report is displayed on a screen, SQL\*ReportWriter is automatically in browse mode. This means that you can scroll from page to page through the report output. You must use Window mode (see below) to examine the parts of the page that do not fit on the screen. When you are done browsing through your output, press [Accept] or [Undo].

### **Window Mode**

Window mode allows you to scroll through your output line by line and character by character. To invoke this mode, press [window]. To return to browse mode, press [Accept] or [Undo].

## DUMPREP

The DUMPREP command copies report definitions from the database into an ASCII file with a default extension of .rex. The .rex file can be moved to other computer systems with any file transfer program that supports text files. DUMPREP is used in conjunction with the LOADREP command to load the ASCII file into another ORACLE database. DUMPREP can also be used along with the GENREP command to generate compiled reports directly from the .rex file.

**syntax** dumprep [REPORTS=] reports [ [USERID=] userid [ [FILE=] file]

where

**REPORTS** the SQL\*ReportWriter report name expression. If you are using System-owned SQL\*ReportWriter tables, you may list reports of other users, separated by commas, for example,

REPORTS= (SDAY . QANDA, DSMITH . TEST, DMAST . SAL)

When using user-owned SQL\*ReportWriter tables, reports from multiple users must be unloaded from each user individually.

**Note:** You may dump only the reports to which you have been granted access. If no userid is prefixed to the report name, the userid is assumed to be the current user. If only one report is being dumped, the parentheses are optional.

Report names may contain the SQL wildcard characters % and\_. All reports that match are dumped.

**USERID** the ORACLE userid/password@node. If you omit the username and/or password, you will be prompted for the missing item(s). To invoke a default logon procedure, using a username prefix With OPSS\$, type dumprep <reports> /on the command line. If this procedure is unsuccessful, you will be prompted for your username and password.

**FILE** the location (path) and name of the load file (the default is expdat.rex)

**Rules** If any of the reports has an invalid query, DUMPREP will place the query in the .rex file with a warning that the report in the .rex file cannot be used with GENREP FILE= <VALUE>. **Note:** Although the .rex files are readable, do not edit them edited .rex files are not supported.



## LOADREP

The LOADREP command loads a report export file into an ORACLE database. The default file extension is .rex. The .rex file is an ASCII file and can be moved with any file transfer mechanism that supports ASCII files. See the DUMPREP command earlier in this chapter for details on .rex files.

Syntax loadrep [ [FILE=] file ] [ [USERID=] userid ]

where:

**FILE** the location (path) and name of the load file (the default is *expdat.rex*)

**USERID** the ORACLE userid /password@node where you want to load the file. If you omit your name and /or password, you will be prompted for both your name and password. To invoke a default logon procedure (using a username prefixed with OPS\$), type loadrep filename /on the command line. If this procedure is unsuccessful, you are prompted for your username and password.

**Rules** LOADREP loads all reports, including those with invalid queries.

## PRINTDEF

The PRINTDEF command is used to create a new printer definition. Execute this command after inserting printer control sequences in the printdef.dat file.

Syntax printdef [ [PRINTER=] printer ] [ [OUTFILE=] out file ] [ [PDFILE=] pdfile ]

where:

**PRINTER** the name of the printer

**OUTFILE** the location (path) and name of the file to place the new printer definition

**PDFILE** the location and name of the printer definition file (the default is printdef.dat). See "Creating/Modifying the Printer Definition File" in Appendix H for more information.

## TERMDEF

The TERMDEF command is used to create a new terminal definition. Execute this command after inserting terminal control sequences in the termdef.dat file. See Appendix G for more information.

**Syntax** termdef [TERMINAL=] terminal\_name [OUTFILE=] out file [ INFILE=] infile [USERID=]userid [[ TDFILE=]tdfile]

where:

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| TERMINAL_ NAME | the second name you entered in the terminal names line                                 |
| OUTFILE        | the location (path) and name of the file in which to place the new terminal definition |
| INFILE         | the location and name of the key definition file                                       |
| USERID         | the ORACLE userid                                                                      |
| TDFILE         | the location and name of the terminal description file (the default is termdef.dat).   |

The arguments must be entered in the order specified above.

## MOVEREP

MOVEREP is the SQL\*ReportWriter Version 1.0 to 1.1 conversion program. To convert your reports, you need the following

- When using ORACLE RDBMS Version 6.0 with the transaction processing option, you need SQL\*ReportWriter Version 1.0 tables in your account, and SQL\*ReportWriter Version 1.1 tables in the SYSTEM account (and the Database Administrator must have granted you access to it).
- When using any other version of ORACLE, you need SQL\*ReportWriter Version 1.0 tables and SQL\*ReportWriter Version 1.1 tables in your account.

**Syntax** moverep [USER ID=] userid

where:

|        |                                                                 |
|--------|-----------------------------------------------------------------|
| USERID | the ORACLE username/password@node of your (not SYSTEM) account. |
|--------|-----------------------------------------------------------------|

Note: When moving to a new release of SQL\*ReportWriter (e.g., from Version 1.1.8 to Version 1.1.10), you may need to regenerate your rep files.

PART



*II*

**REFERENCE**

# 3

## ACTION MENU

**T**his chapter discuss the Action Menu, which is the first choice on the Main Menu. This Action Menu contains a list of options which operate on report definitions, including

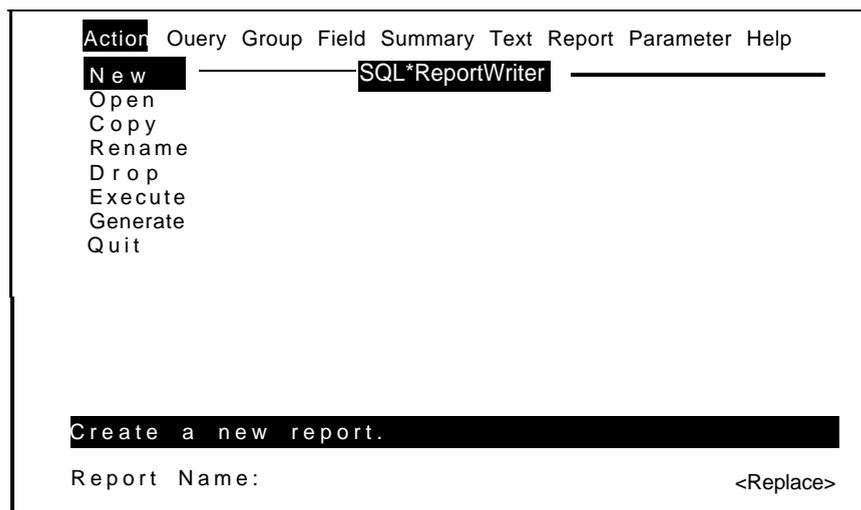
- N e w
- Open
- Copy
- Rename
- Drop
- Execute
- Generate
- Quit

## Action Options

Each action, except Quit, has an associated dialog box for specifying the report definition to work on. Figure 3-1 shows the Action Menu.

- Options
- New creates a new report definition.
  - Open opens an existing report definition for editing.
  - Copy creates a new report by copying an existing report.
  - Rename changes the name of an existing report definition.
  - Drop deletes the specified report definition from the report catalog.
  - Execute generates and runs an existing report, and creates report output.
  - Generate creates a runfile for an existing report.
  - Quit exits to the operating system.

**FIGURE 3-1**  
The Action Menu



## New

Creates a new report definition. The default margins are set on the Report Screen, default Report and Page text objects are created (black) on the Text Screen, and system parameters for DESTYPE, DESNAME, DESFORMAT, COPIES, CURRENCY, THOUSANDS, and DECIMAL are automatically created on the Parameter Screen.

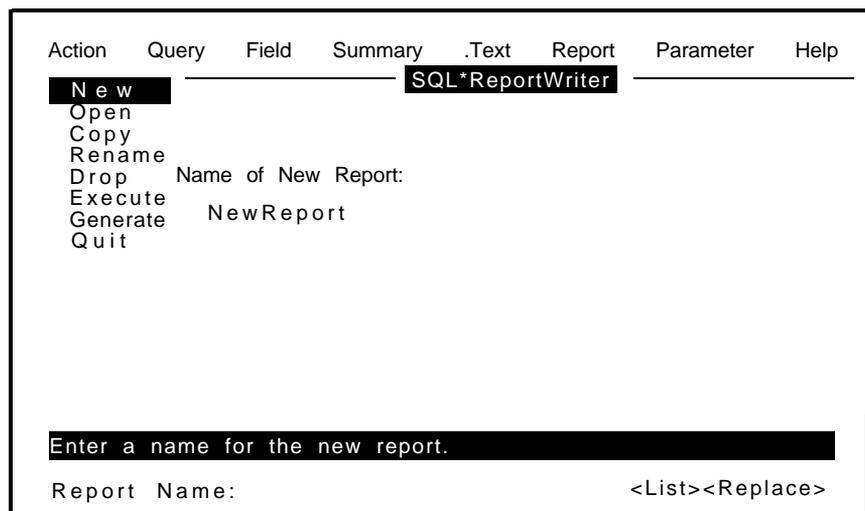
**Default** None.

**options** Enter a name for the report. You can check the List of values for a list of your own reports. In this context this is actually the list of invalid values, i.e., report names you cannot use for new reports.

- Rules**
1. An error occurs if you specify the name of an existing report.
  2. Report names must follow standard SQL naming conventions. Single and double quoted names are not supported.
  3. Once you exit this dialog box, the report is created and exists until you delete (Drop) it.

Figure 3-2 shows the New option selected from the Action Menu.

**FIGURE 3-2**  
Create a New Report



## Open

Opens an existing report definition for editing.

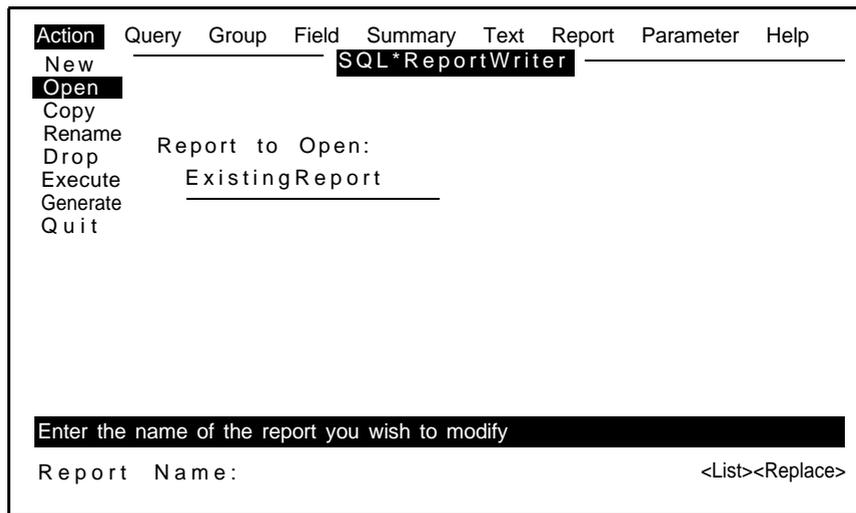
Default None.

Options Enter the name of the report to open by either typing it in or choosing it from List of values, which displays the names of your own reports.

Rules The report must already exist.

Figure 3-3 shows the Open option selected from the Action Menu.

**FIGURE 3-3**  
Open an Existing Report



If you are using the ORACLE transaction processing option, you can open someone else's report to which you have been granted access. To do so, enter the person's username, a period, and then the report name, for example, SMITH.STATUS\_REPORT

## Copy

Creates a new report by copying an existing report definition.

**Default** The open report, if any.

**Options**

1. Choose a report to copy from the List of values, which displays your own report; for both the From and To entries. Note that the List of values for the To contains a list of invalid names, i.e., names you cannot use.

2. Copy a report from another username by entering the following as the report to copy from:

Username. report name

**Rules**

1. The new report is saved when you exit the dialog box.

2. If report definitions are stored in system-owned SQL\*ReportWriter tables, that user must grant you access to the report (on the Report Setting Screen) before you may copy it. See Chapter 11, "Access" rules.

If report definitions are stored in user-owned SQL\*ReportWriter tables, that user must have granted you access to the report by running the srw\_grnt.sql script (using SQL\*Plus). **Note:** With the script, the creator of the report may only grant access to Public (all users) or to a specific person. To grant access to two specific people, script must be run twice, once for each person.

Figure 3-4 shows the Copy option selected from the Action Menu.

FIGURE 3-4  
Copy an Existing Report

The screenshot shows a dialog box titled "Action" with a menu of options: New, Open, Copy, Rename, Drop, Execute, Generate, and Quit. The "Copy" option is selected. To the right of the menu, there are two input fields: "Report to Copy From:" with "OldReport" entered, and "Report to Copy To:" with "NewReport" entered. Below the menu, there is a text prompt: "Enter the name of the report you wish to copy." and a text input field labeled "Report Name:" with a "<List><Replace>" button to its right.

## Rename

Changes the name of an existing report definition.

Default The Old Name is the open report name, if any.

Options Choose a report to rename from the List of values, which shows a list of your reports.

- Rules
1. The Old Name must exist, and the New Name must not.
  2. The name change is permanent when you exit this screen.
  3. You cannot rename someone else's report, even if they have granted you access to their *report*.

Figure 3-5 shows the Rename option selected from the Action Menu.

**FIGURE 3-5**  
**Rename an Existing Report**

| Action        | Query | Group | Field | Summary             | Text | Report           | Parameter | Help |
|---------------|-------|-------|-------|---------------------|------|------------------|-----------|------|
| New           |       |       |       |                     |      | SQL*ReportWriter |           |      |
| Open          |       |       |       |                     |      |                  |           |      |
| Copy          |       |       |       |                     |      |                  |           |      |
| <b>Rename</b> |       |       |       | Old Name of Report: |      |                  |           |      |
| Drop          |       |       |       | OldName             |      |                  |           |      |
| Execute       |       |       |       |                     |      |                  |           |      |
| Generate      |       |       |       | New Name of Report: |      |                  |           |      |
| Quit          |       |       |       | NewName             |      |                  |           |      |

Enter the current name of the report you wish to rename.

Report Name: <List><Replace>

## Drop

Deletes the specified report definition.

Default The open report, if any.

Options Choose the report to drop from the List of values, which shows a list of your reports.

- Rules
1. The report is deleted when you exit the screen.
  2. You cannot drop someone else's report, even if they have granted you access to their report.

Figure 3-6 shows the Drop option selected from the Action Menu.

**FIGURE 3-6**  
**Drop an Existing Report**

| Action      | Query | Field | Summary         | Text             | Report | Parameter | Help |
|-------------|-------|-------|-----------------|------------------|--------|-----------|------|
| New         |       |       |                 | SQL*ReportWriter |        |           |      |
| Open        |       |       |                 |                  |        |           |      |
| Copy        |       |       |                 |                  |        |           |      |
| Rename      |       |       |                 |                  |        |           |      |
| <b>Drop</b> |       |       | Report to Drop: |                  |        |           |      |
| Execute     |       |       | OldReport       |                  |        |           |      |
| Generate    |       |       |                 |                  |        |           |      |
| Quit        |       |       |                 |                  |        |           |      |

Enter the name of the report you wish to delete.

Report Name: <List><Replace>

## Execute

Generates and runs an existing report.

Default The open report, if any.

- Options
1. Choose the report to execute from the List of values, which shows names of the reports that you own. You can execute someone else's report to which you have been granted access. (See "Copy" earlier in this chapter.) To do so, enter the person's username, a period, and then the report name.
  2. To store the runfile in a different directory than is specified in the dialog box, type in a new directory path.
  3. The Run-time Parameter Form appears after Execute is selected, unless you specified `PARAMFORM=NO` on the `SQLREP` command line, or you entered `x` in the Skip column for every parameter. Run the report with the default values, or override them using this Form by entering new values.

- Rules
1. Compilation errors or run-time errors, or both, may occur at this point and will result in error messages. To get help on these messages, press [Help].
  2. Before SQL\*ReportWriter begins to fetch data, it generates a "runfile" that contains a compressed form of the report definition. The runfile has the same name as the report with an extension of `.rep`.
  3. SQL\*ReportWriter does not notify you if another file with that name already exists; if a file does exist, the behavior is operating system dependent. On most operating systems, it will replace the file. Figure 3-7 shows the Execute option selected.

FIGURE 3-7  
Run an Existing Report

| Action   | Group | Field              | Summary | Text | Report           | Parameter | Help |
|----------|-------|--------------------|---------|------|------------------|-----------|------|
| New      |       |                    |         |      | SQL*ReportWriter |           |      |
| Open     |       |                    |         |      |                  |           |      |
| Copy     |       | Directory:         |         |      |                  |           |      |
| Rename   |       | /your/path         |         |      |                  |           |      |
| Drop     |       |                    |         |      |                  |           |      |
| Execute  |       | Report to Execute: |         |      |                  |           |      |
| Generate |       | YourReport         |         |      |                  |           |      |
| Quit     |       |                    |         |      |                  |           |      |

Enter the name of the report you wish to execute.

Report Name: <List><Replace>

## Generate

Creates a runfile for an existing report.

Default The open report, if any.

Options Choose the report to generate from the List of values, which shows a list of reports you own. When accessing data from system-owned SQL\*ReportWriter tables, you can generate someone else's report if you have been granted access to it. (See "Copy" earlier in this chapter.) To do so, enter the person's username, a period, and then the report name.

- Rules
1. Compilation errors or run-time errors, or both, may occur at this point and will result in error messages. To get help on these messages, press [Help]
  2. The runfile has the same name as the report, with an extension of .rep. SQL\*ReportWriter does not notify you if another file with that name already exists; if a file does exist, SQL\*ReportWriter's behavior is operating system dependent. On most operating systems, it will replace the file. See the ORACLE *Installation and User's/ Guide* for your system.
  3. To store the runfile in a different directory than is specified in the -dialog box, enter a new directory path.

Figure 3-8 shows the Generate option selected from the Action menu.

**FIGURE 3-8**  
**Generate a Runfile**

| Action   | Query | Group               | Field | Summary | Text | Report           | Parameter | Help |
|----------|-------|---------------------|-------|---------|------|------------------|-----------|------|
| New      |       |                     |       |         |      | SQL*ReportWriter |           |      |
| Open     |       |                     |       |         |      |                  |           |      |
| Copy     |       |                     |       |         |      |                  |           |      |
| Rename   |       | Directory:          |       |         |      |                  |           |      |
| Drop     |       | /your/path          |       |         |      |                  |           |      |
| Execute  |       |                     |       |         |      |                  |           |      |
| Generate |       | Report to Generate: |       |         |      |                  |           |      |
| Quit     |       | YourReport          |       |         |      |                  |           |      |

Enter the name of the report you wish to generate.

Report Name: <List><Replace>

**Quit**

Exits to the operating system.

Default None.

Options None.

Rules SQL\*ReportWriter does not prompt you to save your work. This is because your work is automatically saved each time you press [Accept] or move between different SQL\*ReportWriter screens.

CHAPTER

# 4

## QUERY SCREEN

**T**his chapter discusses the Query Screen. Queries define the data retrieved from the database. Each query consists of:

- a query name
- a SQL SELECT statement
- parent-child relationships.

## Query Management

Before any queries have been defined, the Query Screen is empty.

### Inserting a Query

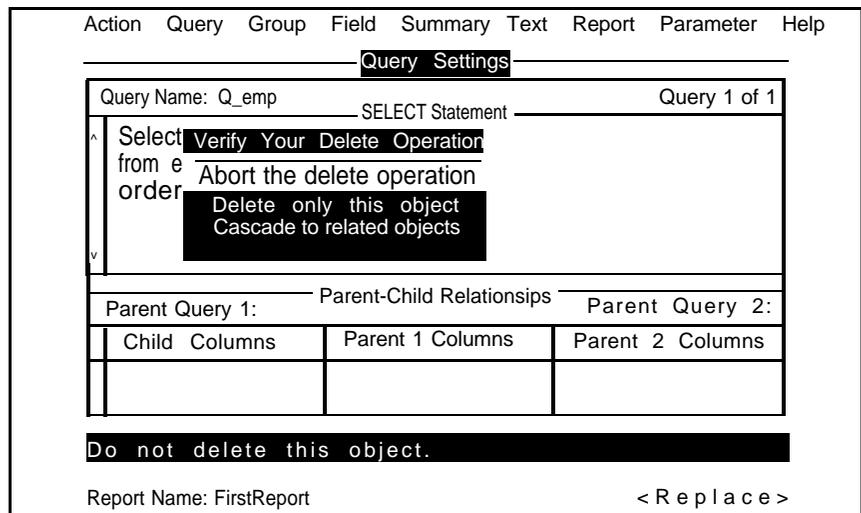
To insert the first query, enter a name for the query in the Query Name entry area, press [Next Field], and then enter the query in the scrollable, multi-line entry area labelled SELECT Statement.

To insert an additional query, press [Insert Record Below] or [Insert Record Above] and repeat the above procedure. Inserting a query between two existing queries will not cause the new query's fields to appear between the fields that belong to the existing queries (on the Field Screens). You must move the new query's group between the existing queries' groups to place the new fields between the existing fields.

### Deleting a Query

To delete a query, press [Next Record] or [Previous Record] until you reach the query you want to delete, then press [Delete Record] to delete the query. An alert box appears with a set of options. See Figure 4-1.

**FIGURE 4-1**  
The Alert Box



- Abort the delete operation cancels the delete.
- Delete only this object deletes the query but does not modify any references to it.
- Cascade to related objects deletes the query and all objects which refer to it.

If you choose the "Delete only..." option,, all references to the query will be labelled \*undefined\*. You cannot run the report until you enter a valid query name for these references.

If you choose the "Cascade..." option, the following objects will also be deleted:

- fields that are derived from query
- field that belong to a group associated with the query
- groups that are associated with the query
- summaries whose print groups are associated with the query
- summaries based on fields which belong to a group associated with the query
- text objects of a group that is associated with the query

## Viewing a Query

To see each of the queries in your report, place the cursor in the query Name entry area and press [Next Record] or [Previous Record].



## Default Objects

To enable you to run a report after specifying only one query, SQL\*ReportWriter generates several default objects for each query

- one field for each column in the SELECT Statement; the width and data type are taken from the underlying data dictionary entries
- default field labels for each field (See "Field Label" for more information)
- one group that contains all fields from the query
- default column heading text for each group; this text contains the field labels for the fields in the group
- default text for the body of each group; this text contains references to each of the fields in the group
- default parameters (see "Default Parameters" in Chapter 10).

The following attributes appear on the Query Screen (Figure 4-2):

- query name
- SELECT Statement
- parent query 1 and parent query 2
- child query columns
- parent 1 columns and parent 2 columns.

**FIGURE 4-2**  
Query Screen

| Action                       | Query | Group            | Field            | Summary | Text | Report           | Parameter    | Help |
|------------------------------|-------|------------------|------------------|---------|------|------------------|--------------|------|
| <b>Query Settings</b>        |       |                  |                  |         |      |                  |              |      |
| Query Name:                  |       | SELECT statement |                  |         |      |                  | Query 1 of 1 |      |
|                              |       |                  |                  |         |      |                  |              |      |
| Parent-Child Relationships   |       |                  |                  |         |      |                  |              |      |
| Parent Query 1:              |       |                  | Parent Query 2:  |         |      |                  |              |      |
| Child Columns                |       |                  | Parent 1 Columns |         |      | Parent 2 Columns |              |      |
|                              |       |                  |                  |         |      |                  |              |      |
| Enter a name for this query. |       |                  |                  |         |      |                  |              |      |
| Report Name:                 |       |                  |                  |         |      |                  | <Replace>    |      |

## Query Name

Specifies the name of the current query. Queries can also be renamed in this entry area. It is a good idea when naming queries to use a standard naming convention.

To the right of the query name a display shows the number of the current query out of the total number of queries (e.g., Query 1 of 1) entered in the report.

**Default** The query name setting is blank for a new query. If there are existing queries, the name of the first query in the report appears in this entry area when you enter the screen.

**Options** Enter a unique name for the query.

- Rules**
1. Query names must follow the SQL naming standards.
  2. Query names must be unique; they cannot have the same name as any other object in the report.

## **SELECT Statement**

Contains one SQL SELECT Statement.

There are two Lists of values that are available in the SELECT Statement area. The first List of values displays all tables from which you may choose. Invoke this list by pressing [List] once. You may view the list, move your cursor to a table name and press [Select] to insert it in your SELECT Statement, or you may use the table list to invoke a second list: a list of all columns for a table in which you are interested. To exit the tables list, press [Undo].

To invoke the list of columns for a particular table, move the cursor to a table name and press [List]. You may simply view the list, or you may move your cursor to a column you want to insert into your SELECT Statement and press [Select]. The column will be inserted where your cursor was in your SELECT Statement. You may select any number of columns from any number of tables from the lists. To exit the columns list, press [Undo]. The SELECT Statement area on the Query Screen is the only place where one List invokes another List.

Default None.

Options Enter a SELECT statement; all features of the SELECT Statement are supported, except the FOR UPDATE clause.

- Rules
1. SELECT Statements must follow all syntax rules governing SQL SELECT Statements.
  2. No semicolon, slash, or any other character should be used to end the SELECT Statement.
  3. The SELECT statement can contain up to 32K characters, including newlines and spaces.
  4. You must have SELECT privileges on the tables and views referenced in the SELECT Statement when you define the report in order for the SELECT Statement to be validated.
  5. SQL\*ReportWriter checks the syntax when the user leaves the SELECT Statement area and when the report is generated, executed, or opened.
  6. If you change a field in your query which causes the width of the field to change, and the field has an alias, the change in width will not be reflected on the Field Screen. You must manually change the Field Width specified on the Field Screen.
  7. If a DATE parameter is used in an expression, a TO\_DATE function should be Used.

8. If SELECT expressions are added to, renamed in, or deleted from a query, SQL\*ReportWriter automatically creates or deletes the corresponding fields. Note: Edited text objects will not reflect your additions or deletions; you must adjust these manually. (For renamings moving or deleting parameters from queries, see Chapter 10.)
9. Fields and summaries based on other queries can be referenced in queries.
10. Fields and summaries that you manually create, by inserting them on the Field Settings or Summary Settings screens, can be referenced in queries. You reference the fields in your SELECT statement by entering a colon and then the field name (no spaces between).
11. You cannot reference a field or summary in a query whose Reset Group is *Page*.
12. You cannot reference a field in a query whose Source is &PAGE or &NUM\_PAGES.
13. A query's field or summary reference must be in a group above the query and cannot be computed from any field appearing in a group in, to the right of, or below the group associated with the query.
14. Parameters can be used to replace a value, or values, in a SELECT statement. There are two types of parameters bind and lexical.

With bind parameters, one value is substituted into the parameter reference; with lexical parameters several values may be substitute into the parameter reference. Thus, bind parameters may be used anywhere in the query where a single literal value, such as a character string number, or date could be used. Lexical parameters can be used in the WHERE, GROUP BY, ORDER BY, HAVING, CONNECT BY, and START WITH clauses, and may replace values as well as SQL expressions.

A default definition is provided for each bind parameter if it has not been created manually. Thus, you can create a bind parameter just by entering a colon and then a parameter name (no spaces between) in your SELECT statement.

A default definition is not provided for lexical parameters. You must, therefore, first define each lexical parameter on the Parameter Screen before referencing it in your query. (Alternatively, you can build your query so that it will be correctly validated if the reference to the lexical parameter is replaced with NULL (nothing). You reference lexical parameters in a query by preceding the name of the parameter with an ampersand (&).

See Chapter 10, "Parameter Screen" for more information about parameters. Examples of bind and lexical parameters are provided below.

15. Changes made to the database will not be recognized by SQL\*ReportWriter automatically. After a change has been made, you must change the Source (on the Field Screens) for the offending fields, and then change the Source back again.
16. Parameter names must be distinct from all other report objects, RUNREP command line keywords (see Chapter 2, "RUNREP"), and they must conform to SQL naming standards. Bind parameters must not be the same as any reserved SQL keywords. For a list of reserved SQL keywords, consult the SQL Language Reference Manual.
17. Bind parameters may be used in the following SELECT statement clauses: SELECT, WHERE, GROUP BY, ORDER BY, CONNECT BY, HAVING, and START WITH. Bind parameters may not be referenced in FROM clauses or in place of reserved words or clauses. The following are examples of valid uses of bind parameters:

```
DEPTNO = : DEPT (after WHERE)
```

```
SELECT NVL (DEPTNO, : DEPT)
```

The following are examples of invalid uses of bind parameters:

```
FROM : MYTAB (bind parameters can not be used in a FROM clause)
```

```
WHERE : MYTEST (WHERE must compare two values, bind parameters
replace one)
```

18. After lexical parameters have been defined and given default values, they may be used to replace all or part of the following SELECT statement clauses: WHERE, GROUP BY, ORDER BY, CONNECT BY, HAVING, and START WITH. Lexical parameters may not be referenced in SELECT or FROM clauses or in place of SQL reserved words or clauses. The following is an example of a valid use of lexical parameters:

```
SELECT ENAME FROM EMP
WHERE &MYPARAM
```

The following is an example of an invalid use of lexical parameters:

```
SELECT &MYCOLUMN
FROM EMP (lexical parameters can not be used in a
SELECT' clause)
```

19. The order of the fields in the SELECT statement determines the initial field order. Changing the order of columns in the SELECT statement will not change the order of the fields. If you add a new field to the SELECT statement, the corresponding field will be added at the end of the bottom-most group created by the query. See Chapter 6, "Field Screens," to see how to change field order.
20. SELECT statements specifying duplicate column names from different tables are not supported. If necessary, use an alias for one of the column names.

## Parent-Child Relationships

These settings allow you to relate the results of multiple queries. When you specify a parent query and pairs of matching columns, SQL\*ReportWriter retrieves only the rows in the child query that match the rows in the parent. The child query is re-executed for each new row retrieved by the parent query. When queries are related, a field can be created in a lower group which is a copy of a field from a higher group, or a field in a higher group can be a summary of a field in a lower group.

All rows in the parent query, or queries, are displayed whether or not any matching rows are found in the child query. Note: In a Master-Detail report, if a parent is not printed on a page, its text is compressed to take up no room, but its Lines Before and Spaces Before are still formatted. To avoid this "inheritance," add carriage returns and spaces to the desired text object to achieve the same effect as the Lines Before and Spaces Before Group settings.

### Parent Query 1

Specifies a query to be the parent of the query displayed on the screen.

Default None.

Options Choose a parent query from the List of values, which shows a list of names of potential parent queries.

- Rules
1. No query can be its own parent, directly or indirectly.
  2. Parent Query 1 must be different from Parent Query 2.

### Parent Query 2

This setting is used only for matrix reports. It provides the second dimension of the matrix. See Matrix in the "Group Screens" chapter.

Options Choose a second parent query from the List of values, which shows a list of names of potential queries.

Default None.



CHAPTER

# 5

## GROUP SCREENS

**T**his section discusses the three Group Screens. A group is a set of one or more fields, and is used to organize report formats and define master/detail relationships. A group is one of the objects that comprises a SQL\*ReportWriter report.

## Group Management

Groups, like other report objects, can be inserted, deleted, moved, and renamed. A group must be created on the Group Screens before it can be referenced elsewhere in the report definition.

SQL\*ReportWriter automatically creates a group for each query you enter. These groups are named by attaching a "G\_" to the query name. The only exception is when query names start with a "Q". In that case, SQL\*ReportWriter removes the "Q" and replaces it with "G\_" when naming the associated group.

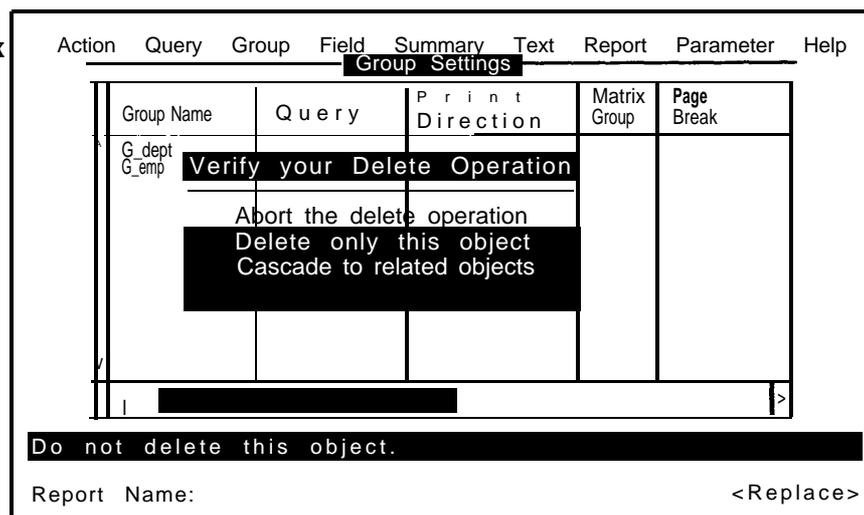
### Inserting a Group

To insert a group between other groups, use [Insert Record Above] or [Insert Record Below] to open a blank line above or below the group that the cursor is on. You can also add a group by inserting a line below the last group in the list. After you enter a Group Name, you must also assign that group to a Query. lastly, you must move to the Field Settings Screen and assign at least one field to the group. (The advantage of being able to insert groups is that you can create control breaks using the fields in existing queries.) When there is more than one group assigned to a query, a field can be created in a lower group which is a copy of a field from a higher group, or a field in a higher group can be a summary of a field in a lower group.

### Deleting a Group

To delete a group, position the cursor on the group and use [Delete Record]. when a group is deleted, an alert box (Figure 5-1) appears with a list of options, which are described below. Select the appropriate option.

**FIGURE 5-1**  
**Delete Operations Alert Box**



- Abort the delete operation cancels the delete.
- Delete only this object puts the group in the delete buffer and does not modify any references to it.
- Cascade to related objects deletes the group and all objects which refer to it.

If you choose the “Delete only...” option, all objects which reference the group will be labelled “Undefined”. You cannot run a report that has a reference to an undefined object. Either undelete the deleted group or replace the undefined references with valid values.

If you choose the “Cascade...” option, the following objects will also be deleted:

- Fields that belong to the group
- Summaries which print at the deleted group
- Summaries based on fields which belong to the group
- Text objects that are associated with the group.

## **Moving a Group**

Moving a group above (or below) another group causes all fields owned by the group to move above (or below) the fields owned by the stationary groups. To move a group to a different position, do the following

1. Move the cursor to the group and press [Delete Record].
2. Choose “Delete only...” so that current references to fields are preserved.
3. Position the cursor on the group that you want to follow the deleted group and press [Undelete Record].

## **Renaming a Group**

To rename a group, type the new name over the existing name on Group Screen One.

## **Ordering Groups**

You can arrange groups in any order on the Group Screens. When you exit this screen, SQL\*ReportWriter re-sorts them according to the following rules, if necessary:

1. Groups always appear in the order implied by the hierarchy of the associated queries.
2. If one query is defined as the parent of another, the groups that reference the parent will appear before the groups that reference the child.

3. All the groups that belong to the same query appear together.
4. Groups that belong to the same query can appear in any order.

## Group Screen One

Group Screen One (Figure 5-2) is used to specify:

- the name of each group
- the query each group belongs to
- the direction in which successive rows in the group print
- matrix groups
- page break conditions.

**FIGURE 5-2**  
Group Screen One

| Action                      | Query      | Group | Field           | Summary      | Text       | Report | Parameter | Help |
|-----------------------------|------------|-------|-----------------|--------------|------------|--------|-----------|------|
| <b>Group Settings</b>       |            |       |                 |              |            |        |           |      |
|                             | Group Name | Query | Print Direction | Matrix Group | Text Break |        |           |      |
|                             |            |       |                 |              |            |        |           |      |
| Enter a name for this group |            |       |                 |              |            |        |           |      |
| Report Name:                |            |       |                 |              |            |        | <Replace> |      |

### Group Name

Specifies the name of the group.

**Default** The name of the query, prefixed with G\_; if the Query Name begins with Q it is replaced by G\_. If the Query Name does not begin with Q it is given the prefix of G\_. There is no default name for groups you create manually.

**Options** Create new groups by inserting records, entering new group names, and assigning fields to them. A prefix of G\_ is recommended, but not required.

**Rules** 1. Group names must be distinct from each other and from query, field, summary, and parameter names, as well as the report's name.

2. Define groups on this screen before assigning the fields to them.
3. You must assign at least one field to each group in the report definition. This restriction is validated when you execute or generate the report.
4. There is no limit to the number of groups that can be created.

## Query

Specifies the query with which the group is associated.

**Default** For default groups, the query from which they were created. For manually created groups, none.

**Options** Choose a query from the List of values, which shows the list of valid query names.

**Rules** The query you enter must already exist.

## Print Direction

Specifies the direction in which successive records of the group appear.

**Default** Down.

**Options** Down prints each record of the group below the previous record.

Across prints each record of the group to the right of the previous record.

Down/Across prints each record of the group below the previous record until the bottom of the page is reached. At that time, SQL\*ReportWriter prints the record to the right of the top-most record, provided there is still adequate room on the page.

Across/Down prints each record of the group to the right of the previous record until the right margin of the page is reached. At that time, SQL\*ReportWriter prints the record below the left-most record on the same page, provided there is still adequate room on the page.

Crosstab prints each record of the group as a cell in the matrix.

- Rules**
1. A single report cannot mix *Down* and *Across* groups, except in a matrix report. Therefore, a non-matrix report must consist only of Down and Down/Across or of Across and Across/Down groups.
  2. See the Matrix setting for rules relating to matrix reports.
  3. Print Direction affects the default placement of fields. When the Print Direction is Down, Down/Across, or Crosstab, each field is placed to the right of the preceding field. When the Print Direction is Across or Across/Down, the fields are placed below each other.
  4. Wrap and Variable fields are not supported in groups with a Print Direction of *Across* or *Across/Down*.

5. If a group is to the right of a Down/Across group or below an Across/Down group, records will overlap unless you manually spare the group far enough away from the previous group.

## **Matrix Group**

Specifies that the indicated group is part of a matrix report. You define a matrix (or crosstab) report by using three groups and selecting the Matrix option for each. The first two groups form the two dimensions of the matrix; the third forms the cells.

Default Blank, which indicates that the group is not a matrix group.

Options X indicates that the group is a matrix group.

- Rules
1. If any groups are marked as Matrix, then there must be exactly three groups in your report and they must all be marked Matrix.
  2. Each group must be associated with a different query. The query associated with the Crosstab group must have the other two queries as its parents (specified as Parent 1 and Parent 2 fields on the Query Screen).
  3. The three matrix groups in a report must be the only groups in the report.
  4. The Print Direction of one matrix group must be *Across*, the second must be *Down*, and the third group must be *Crosstab*.
  5. Page Break and Multi-panel settings are not supported for matrix groups.
  6. Wrap and Variable fields are not supported in matrix reports.
  7. The ORDER BY clause of all three queries must be consistent. Columns in the queries forming the Down and Across groups must appear in the same order in the query for the Crosstab group.

## **Page Break**

Allows pagination control for each group.

Default Blank, which implies a page break when the current page is full.

Options Always causes SQL\*ReportWriter to insert a page break between each record in the group (starting between the first and second record).

Conditional causes SQL\*ReportWriter to attempt to place all output from a single record of that group on one page, including the column header (if outputting the first record), the record data, and all output from any child groups. If any of this output cannot fit on the current page, SQL\*ReportWriter places all of it on the subsequent page. If the output is too large to fit on a single page, SQL\*ReportWriter places the output at the top of the next page and overflows it as necessary.

## Group Screen Two

The following attributes are specified on Group Screen Two (Figure 5-3):

- the relative position of each group with respect to the previous group
- the number of lines and spaces to skip before printing records in the group
- the number of lines and spaces between fields and records within the group
- the maximum number of fields that will print across a panel.

**FIGURE 5-3**  
**Group Screen Two**

| Action Query Group Field Summary Text Report Parameter Help |                   |              |               |                |       |               |
|-------------------------------------------------------------|-------------------|--------------|---------------|----------------|-------|---------------|
| Group Settings                                              |                   |              |               |                |       |               |
| Group Name                                                  | Relative Position | Lines Before | Spaces Before | Record Spacing | Field | Fields Across |
|                                                             |                   |              |               |                |       |               |

< | >

**Choose the position of this group in relation to the previous group.**

Report Name: <List><Replace>

### Relative Position

Specifies the position of the group in relation to the previous group.

- Default** Blank, which implies Right for Down and Down/Across groups and Below for Across and Across/Down groups.
- Options** Right prints the group to the right of the previous group.  
Below prints the group below the previous group.
- Rules**
1. Below causes the group to print at the left margin of the prior group if the print direction is Down or Down/Across.
  2. Right causes the group to print at the top margin of the prior group if the print direction is Across or Across/Down.

## **Lines Before**

Specifies the number of blank lines before a group relative to the prior group or the top margin.

Default Blank, which implies 2 for the second and subsequent Across and Across/Down groups, and 0 for all other groups.

Options Any number between 0 and the number of lines remaining before the bottom margin.

Rules 1. In a Master-Detail report, if a parent is not printed on a page, its text is compressed to take up no room, but its Lines Before and Spaces Before are still formatted. To avoid this "inheritance," add carriage returns and spaces to the desired text object to achieve the same effect as the Lines Before and Spaces Before Group settings.

## **Spaces Before**

Specifies the number of blank spaces before a group relative to the prior group or the left margin.

Default Blank, which implies 2 for the second and subsequent Down and Down/Across groups, and 0 for all other groups.

Options Any number between 0 and the number of spaces remaining before the right margin.

Rules 1. In a Master-Detail report, if a parent is not printed on a page, its text is compressed to take up no room, but its Lines Before and Spaces Before are still formatted. To avoid this "inheritance," add carriage returns and spaces to the desired text object to achieve the same effect as the Lines Before and Spaces Before Group settings.

## **Record Spacing**

Specifies the spacing between each record in the group.

Default Blank, which implies:

- 0 lines for Down and Down/Across groups
- 2 spaces for Across and Across/Down groups.

Options Any number greater than or equal to 0.

Rules 1. When the Print Direction is Down or Down/Across, this setting affects lines between records.  
2. When the Print Direction is Across or Across/Down, this setting affects spaces between records.

## **Field Spacing**

Specifies the spacing between each field in the group.

Default Blank, which implies

- 2 spaces for Down and Down/Across groups
- 0 lines for Across and Across/Down groups.

Options Any number greater than 0.

- Rules
1. Override the default spacing for individual fields of Down and Down/Across group by entering a value in the Spaces Before setting on the Field Screens.
  2. Override the default spacing for individual fields of Across and Across/Down groups by entering a value in the Lines Before setting on the Field Screens.
  3. When the Print Direction is Down or Down/Across, this setting affects spaces between fields.
  4. When the Print Direction is Across, or Across/Down, this setting affects lines between fields.

## **Fields Across**

Specifies the maximum number of fields of the group that will print on a single line of a panel for Down and Down/Across groups, or within a single column area of a panel for Across and Across/Down groups. This setting applies to the fields from one record within a single group, not to fields from multiple groups on the same page or to multiple records from the same group. You may override this setting by explicitly positioning individual fields on the Field Screens or by editing the Body of the group on the Text screen (see Rules).

When the Print Direction is Across or Across/Down, this setting affects the number of fields that print until SQL\*ReportWriter reaches the last line of the group. In effect, this setting becomes "Fields Down."

Default Blank, which implies no fixed limit on the number of fields.

Options Any value greater than or equal to 1, or blank.

Rules Override this setting for individual fields in the following ways:

- Specify a Relative Position of Below or Panel for fields that belong to Down and Down/Across groups. SQL\*ReportWriter starts a new line counting from 0 as it does when it runs out of room.
- Specify a Relative Position of Right or Panel for fields that belong to Across and Across/Down groups. SQL\*ReportWriter starts a new column area counting from 0 as it does when it runs out of room.

- Specify Spaces Before on individual fields of Down and Down/Across groups in order to force a new line.
- Specify Lines Before on individual fields of Across and Across/Down groups in order to force a new column area.
- Edit the Body of the group.

## Group Screen Three

Group Screen Three (Figure 5-4) specifies the following layout attributes for groups

- whether to keep all fields in a row on the same panel, by default
- where to place the labels (column headers) for each field in the group
- the default highlighting style for fields and labels.

**FIGURE 5-4**  
Group Screen Three

| Group Settings                                 |             |                |           |       |           |
|------------------------------------------------|-------------|----------------|-----------|-------|-----------|
| Group Name                                     | Multi-Panel | Label Position | Highlight |       |           |
|                                                |             |                | Field     | Label |           |
| ^                                              |             |                |           |       |           |
| v                                              |             |                |           |       |           |
|                                                |             |                | <         |       |           |
| Keep all fields in a record on the same panel. |             |                |           |       |           |
| Report Name:                                   |             |                |           |       | <Replace> |

### Multi-panel

Specifies whether or not to keep all fields in a group on the same panel.

Default Blank, which causes SQL\*ReportWriter to keep all fields of the group on the same panel, if possible.

Options X allows fields to overflow onto multiple panels, without wrapping. The extra fields appear on the same line on each subsequent panel for Down and Down/Across groups, or in the same column area on each subsequent panel for Across and Across/Down groups.

- Rules
1. If multiple groups appear on the same panel and if the body of a higher group has been edited (and a lower group's body has not), fields in the higher group will wrap onto the next panel, leaving room for the edited group.
  2. SQL\*ReportWriter formats each group in the order in which the groups appear on the Group Screen (from the top group to the bottom). Within each group, SQL\*ReportWriter formats each field in the order in which the fields appear on the Field Screens. If any fields in a group are too wide, they are placed on the next panel, regardless of the Multi-panel setting.

**Label Position**

Specifies whether to place the field labels in the Column Heading or Body. This setting works in conjunction with the Print Direction flag.

Default Default behavior is dependent on the Print Direction: Blank means Above for Down and Down/Across groups, and Left for Across and Across/Down groups.

Options Above places the field label above the value of the field.

Left places the field label to the left of the field value.

- Rules
1. If the Print Direction is Down or Down/Across and the Label Position is Above, each label has a row of hyphens below it, and the labels appear in the Column Heading.
  2. If the Print Direction is Down or Down/Across and the Label Position is Left, each label has a space between it and its field, and appears in the Body.
  3. If the Print Direction is Across or Across/Down and the Label Position is Left, each label has at least one space to the right of it, and the labels appear in the Column Heading.
  4. If the Print Direction is Across or Across/Down and the Label Position is Above, the label appears with its field in the Body.
  5. The following table indicates the text object in which the field labels appear.

|                    |             | Label Position |               |
|--------------------|-------------|----------------|---------------|
|                    |             | Left           | Above         |
| Print<br>Direction | Down        | Body           | Column Header |
|                    | Down/Across | Body           | Column Header |
|                    | Across      | Column Header  | Body          |
|                    | Across/Down | Column Header  | Body          |

## Highlight

Use the Field and Label settings to define consistent highlighting for all fields and labels in a group. Override these settings for individual fields by explicitly highlighting areas of the column header or body on the Text Screen (see Chapter 8, “Highlighting Text”). To highlight individual fields and/or pieces of text, see the same section of Chapter 8. To conditionally highlight a field, see “Conditionally Highlighting Fields” in Appendix E.

In addition to highlighting, SQL\*ReportWriter also supports general purpose printer codes. You can embed these codes in your report by editing the text objects. See Chapter 8, “Highlighting Text; for details.

In this context, the *Underline* highlighting attribute means that SQL\*ReportWriter will attempt to use an “underline” attribute, not a row of hyphens beneath the field. Highlighting is not supported for all printers. Check with your System Administrator for details.

## Field

Specifies the highlighting style for the fields in the group. This setting affects all fields in the group. Specify individual highlighting styles or any combination of styles.

Default Blank, which specifies normal typeface.

Options A List of values is provided which displays all options.

|           |                        |
|-----------|------------------------|
| Normal    | Underline/Reverse      |
| Underline | Underline/Bold         |
| Reverse   | Reverse/Bold           |
| Bold      | Underline/Reverse/Bold |

- Rules
1. Highlighting text in this manner is only possible if the text object that contains the group’s fields has not been edited. (See Chapter 8, “Status,” for information on how to identify edited text objects.)
  2. The preceding highlighting styles may not be supported on all output devices. See your System Administrator for information on existing printer definitions, or refer to Appendix H to create your own printer definition.

## Label

Specifies the highlighting style for the field labels in the group. This setting affects all labels in the group.

Default Blank specifies normal typeface.

Options A List of values is provided which displays all options.

|           |                        |
|-----------|------------------------|
| Normal    | Underline/Reverse      |
| Underline | Underline/Bold         |
| Reverse   | Reverse/Bold           |
| Bold      | Underline/Reverse/Bold |

- Rules
1. Highlighting text in this manner is only possible if the text object that contains the group's labels has not been edited. (See Chapter 8, "Status," for information on how to identify edited text objects.)
  2. The preceding highlighting styles may not be supported on all output devices. See your System Administrator for information on existing printer definitions, or refer to Appendix B to define your own printer definition.
  3. If Underline is specified, then the default column heading for that group will not include a row of hyphens below the labels.

CHAPTER

# 6

## FIELD SCREENS

**T**his chapter discusses the three Field Screens. Use these screens to:

- specify attributes for individual fields in a report
- associate user exits with fields
- determine positioning and spacing of fields
- edit field labels
- create computed fields.

## Field Management

Fields, like groups and summaries, can be inserted, deleted, moved, and renamed. By default, each column or expression from each query in a report appears once on these screens. You can replace the *value* from the query with a computed value by supplying a function and reset group on Field Screen Three.

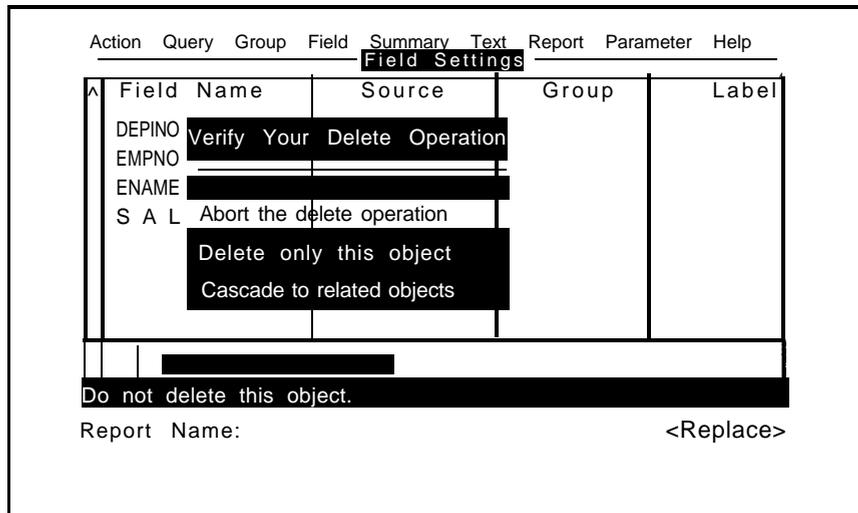
### Inserting a Field

Use [Insert Record Above] or [Insert Record Below] to open a blank line either above or below the line containing the cursor. Enter a name and corresponding information.

### Deleting a Field

Position the cursor on the field and press [Delete Record]. An alert box (Figure 6-1) appears with a list of options

**FIGURE 6-1**  
Delete Operations Alert Box



- Abort the delete operation cancels the delete.
- Delete only this object deletes the field by placing it into the delete buffer but does not modify any references to it.
- Cascade to related objects deletes the field and all objects which refer to it.

If you choose the “Delete...” or “Cascade...” option, any reference to the field that is in in a summary or default text object will also be deleted.

## Moving a Field

To move a field, move the cursor to the field you want to move and press [Delete Record], select the "Delete only this object" option from the alert box, then move the cursor above the field where you want the deleted field to appear, and press [Undelete Record].

## Renaming a Field

To rename a field, type a new name over the existing name. After you do so, all references to that field will also be renamed, except those in edited text objects. To change a field name in edited text objects, type over the field reference (and the field label if desired) in the text object.

## Field Screen One

You use Field Screen One (Figure 6-2) to enter the following information.

- the field name
- the source from which the field is derived
- the group to which the field belongs
- the label to appear as the column heading.

**FIGURE 6-2**  
Field Screen One

The screenshot shows a software interface with a menu bar at the top containing: Action, Query, Group, Field, Summary, Text, Report, Parameter, Help. Below the menu bar is a sub-menu titled "Field Settings". The main area contains a table with four columns: Field Name, Source, Group, and Label. Below the table is a text input field with a cursor and a right arrow. Below the input field is a prompt: "Enter a name for this field." followed by "Report Name:" and "<Replace>".

| Field Name | Source | Group | Label |
|------------|--------|-------|-------|
|            |        |       |       |

Enter a name for this field.  
Report Name: <Replace>

**Field Name**

Specifies the name of each field.

Default SQL\*ReportWriter uses the column name, alias, or text of the source column or expression as the name for default fields, modified as follows

- drops query/table name prefix-Replaces all spaces, parentheses, and operators (+, -, /, \*, | |) in the expression with underscores.
- replaces all duplicate underscores with a single underscore, and drops trailing and leading underscores.
- truncates to 30 characters if the resulting string is greater than 30 characters.
- if the resulting name is not unique, appends a "2" for the second occurrence of the name, "3" for the third, etc.; it will truncate enough characters to ensure that the resulting name is 30 characters or less.

Options This name can be changed without affecting the report definition when the text panel that contains the field has not been edited.

- Rules
1. This name must be distinct from the names of queries, groups, fields, summaries, text objects, parameters, DATE, PAGE, NUM\_PAGES, REPORT, and must follow the SQL naming standards. If field is used as a bind parameter in a query, it cannot be a SQL reserved word.
  2. The order in which the fields appear on this screen is the order in which the fields appear in the report output, except when you edit the text of the group to which the fields belong. See Chapter 8, "k-creating Default Text" for more information.
  3. If the name is changed and the panel that contains the field has been edited, the text panel will not reflect the field name change.

**Source**

Specifies where the field's data comes from

- a column, expression, or alias from the SELECT statement that generates the field
- a system variable (&DATE, &PAGE, and &NUM\_PAGES) that generates the field
- a user exit name and arguments (See Appendix E for more details on how to define user exits.)
- any arbitrary SQL statement preceded by &sql.

## Default

For default fields, the column name, alias, or SQL expression from the query. SQL\*ReportWriter prefixes non-unique column names and aliases with the name of the query. There is no default for system variables, user exits, or printer codes.

## Options

For default fields, choose any of the select expressions from the List of values, which shows the Source columns from all of the queries in the report.

For system variables, enter &DATE, &PAGE, or &NUM\_PAGES in this column. Creating a field for a system variable gives you the flexibility to change their width and format before placing them in your report.

For user exit calls, enter a pound sign (#) and follow it with the user exit name and arguments. A user exit is a subroutine that you write and link into the SQL\*ReportWriter executable. When you call this subroutine from SQL\*ReportWriter, control temporarily passes processing to the user exit. When a user exit has finished executing control is returned to SQL\*ReportWriter.

User exits can:

- perform complex data manipulation
- compute mathematical functions
- pass data to SQL\*ReportWriter from operating system text files
- manipulate LONG RAW data
- execute PL/SQL blocks and SQL commands.

### User Exit Call Syntax

#*userexit* *text*

where:

*userexit* is the pre-packaged user exit name, or the name of your own user exit. The user exit name may be at most 10 characters in length. (On some operating systems the name may be at most 6 characters. Check with your System Administrator.)

*text* can be arguments, printer codes, or constants, or any combination thereof, that you wish to pass to the user exit. The entire string may be at most 240 characters.

An example of a valid user exit call is as follows:

```
#SALARY WORKHOURS COMMISSION VACATION_DAYS BONUS
```

For arbitrary SQL statements, enter an “&sql” (without the quotes), and then enter any SQL statement that returns only a single row. In the SQL statement, identify SQL\*ReportWriter fields that are to be used as bind parameters by preceding them with a colon (:). To put values into SQL\*ReportWriter fields, include an INTO clause in the SQL statement with colons preceding the SQL\*ReportWriter fields. Using a SQL statement as a field’s Source enables you to modify the database without writing a user exit.

Note: The entire text string is passed unmodified to the user exit, and it is the user exit’s responsibility to parse the string. The user exit field is not automatically updated. If you wish to pass a value back to a SQL\*ReportWriter field, summary, or parameter, you must pass that field, summary, or parameter name in the text of the user exit syntax, and then have the user exit set the field value. In general, to read or update a SQL\*ReportWriter field, the user exit must contain an EXEC IAF GET and/or an EXEC IAF PUT ORACLE Precompiled call. See Appendix E for details.

## SQL statement Syntax                      &sql SQL\_Statement

where:

SQL\_Statement            can be any arbitrary SQL statement (up to 240 characters in length) that returns only one row.

Three examples follow:

```
&sql update emp set done=' X' &sql select :a + :b into :c
where ename= :ename from dual
```

```
&sql select ename into :sqlfield from emp
where empno= 10
```

- Rules
1. Any valid Source can be the Source of more than *one* field. (With system variables, for example, this enables you to have different formats of the same system variable in one report).
  2. You cannot make a computed field on a field associated with a user exit, or on a field whose Source is &DATE, &PAGE, &NUM\_PAGES, &sql. If you wish to do so, create a summary of the user exit field on the Summary Screen.

3. If you change a field in your query which causes the width of the field to change, and the field has an alias, the change in width will not be reflected on the Field Screen. You must manually change the Field Width specified on the Field Screen.
4. Changes made to the database will not be recognized by SQL\*ReportWriter automatically. After a change has been made, you must change the Source (on the Field Screens) for the offending fields, and then change the Source back again.
5. If a DATE parameter is used in an expression, a TO\_DATE function should be used (e.g., &sql select 'x' into : flag from dual where to\_date ( : hiredate, 'YYYYMMDDHH24MISS' ) <sysdate).
6. User exits are called in the same order in which fields appear, and are calculated, on this screen: top field to bottom field. Thus, if you have four fields on this screen, and the last one is a user exit, the user exit is called after the first three fields. There is one exception to this rule if a field that is passed to a user exit is below the user exit, that field will be calculated before it is passed to the user exit. Thus, if the eighth field is passed to a user exit (which is the fourth field), the eighth field will be calculated before the user exit.
7. To change the value of a field from within a user exit, the user exit must explicitly PUT a value (i.e., execute an EXEC IAF PUT statement) into the field. If this is not done, the value of the field in the user exit will remain unchanged and the value of the field associated with the user exit will be null.
8. A user exit may not change the value of a field that has a Source of a column from a SELECT statement. If you wish to do so, create a new field with the same Source below the field you want to change, and PUT the new value into the new field.
9. When using &sql, reading updated or inserted values from the database is not advised in the same report, since there is no guarantee of the exact time SQL\*ReportWriter will fetch records from the database for formatting the output. SQL\*ReportWriter builds internal "dependency lists" which guarantee that events, such as invocation of user exits, calculation of summaries, etc., happen in the correct order. However, SQL\*ReportWriter cannot guarantee these events will be synchronized with its internal data access.

Note that SQL statements used in &sql fields are committed immediately after they are executed. This guarantees that your updates/inserts will be made at that time.

SQL\*ReportWriter does internal "data look ahead" to optimize performance. Thus, a particular record might already have been accessed before an update is issued to the same record. You cannot, therefore, read records that you have updated and be guaranteed consistent data.

10. It is guaranteed that &sql will fire sometime after its dependencies have been computed. If there are no dependencies, &sql could fire at anytime. Note: &sql cannot depend on a field that belongs to a group below its group, to a sibling group, or to a group belonging to another parent.

## **Group**

The name of the group to which the field belongs.

**Default** The default group for the associated query from which the field is being retrieved. There is no default for user exits.

**Options** Assign a field to a group by selecting a group from the List of values.

- Rules**
1. Fields are ordered according to the order of their groups. Within groups, fields are ordered by their position on the field screen. By default, the field screen order is initially determined by the order of the columns in the SELECT statement.
  2. A group's query must be in the same path as the query defining the Source.
  3. Copies of fields from a higher group can appear in a lower group, but copies of fields from a lower group cannot appear in a higher group.
  4. A computed field based on a field in a lower group can appear in a higher group, but not vice versa. (See "Computed Value" later in this chapter for information about computed fields.)
  5. A group of REPORT is valid only for fields with a Source of a system variable, or a user exit.

## **Field Label**

Specifies the text of the field label.

**Default** The default field label for a field that is automatically created from a query is the text of the column specified in the SELECT statement, modified following the same rules used for generating field names with the following exceptions:

l spaces are used in place of all non alpha-numeric characters. Labels over 30 characters in length are truncated. There is no special processing to prevent duplicate field labels.

- the first character of a label is capitalized by default, regardless of how the SELECT statement is specified.

Field names that are manually created (entered directly on the Field Screen) have a default label of the entered name with initial caps.

**Options** Change the label by typing over the default or remove it by deleting all the characters.

- Rules**
1. If the label is removed from a field of a Down or *Down/Across* group, and the Label Position of the group is *Above*, then the hyphens that regularly appear beneath the label are also removed.
  2. The label for a field with a datatype PRT is ignored. (See Field Screen Two for details on datatypes.)



## Field Screen Two

Field Screen Two (Figure 6-3) is used to specify positioning and display characteristics for individual fields by entering the following information:

- datatype
- field width
- display format
- relative position
- number of lines and spaces to leave blank before each field.

**FIGURE 63**  
**Field Screen Two**

| Action Query Group Field <u>Summary</u> Text Report Parameter Help |            |           |             |                |                   |              |               |
|--------------------------------------------------------------------|------------|-----------|-------------|----------------|-------------------|--------------|---------------|
| Field Settings                                                     |            |           |             |                |                   |              |               |
|                                                                    | Field Name | Data Type | Field Width | Display Format | Relative Position | Lines Before | Spaces Before |
| A                                                                  |            |           |             |                |                   |              |               |
| V                                                                  |            |           |             |                |                   |              |               |
|                                                                    | <          |           |             |                |                   |              | >             |

**Enter the width of this field in spaces**

Report Name: <Replace>

## Data Type

Lists the datatype for the field. This attribute can be edited when a field's Source is a user exit.

**Default** SQL datatype of the field: *DATE*, *CHAR*, or *NUM*. (For user exits, the default is *CHAR*. See Appendix E for details.)

- Rules**
1. By default, SQL\*ReportWriter defines the datatype for the field based on the datatype of the SELECT expression for the field.
  2. If the Source column is changed, the datatype of the field changes to match, except when the Source is a user exit.
  3. The datatype can be changed manually only when the Source is a user exit. Possible values are *DATE*, *CHAR*, *NUM*, or *PRT* (printer codes). You create a field with a datatype of *PRT* to enable you to conditionally highlight fields, i.e., to put all negative amounts in boldface. For details, see "Conditionally Highlighting Fields" in Appendix E.
  4. Fields with a datatype of *PRT* have a height of 1 and a width of 0.

## Field Width

Specifies the width of each field.

**Default** The width of the Source column derived from the query.

**options** Valid numbers are 0 through 999.

- Rules**
1. If the SQL datatype of the Source column is *LONG*, then the Data Type of the SQL\*ReportWriter field is *CHAR*.
  2. If you change a field in your query which causes the width of the field to change, and the field has an alias, the change in width will not be reflected on the Field Screen. You must manually change the Field Width specified on the Field Screen.
  3. If the value to be displayed is longer than the width of the field, and the datatype is *NUM* or *DATE*, SQL\*ReportWriter replaces the value with asterisks. If the Data Type is *PRT* (a printer code), the field width is 0. If the datatype is *CHAR*, SQL\*ReportWriter will:
    - Wrap the text retrieved within the field width when the field alignment on Field Screen Three is set to *Wrap*
    - Wrap the text retrieved, and fields and boilerplate text following it, within the width of the text object when the field alignment on Field Screen Three is set to *Variable*
    - Truncate the text retrieved in all other cases.

4. If the field does not have Wrap or Variable alignment and the value to be displayed is less than the width of the field, the field will take up the specified Field Width. If the field has Wrap or Variable alignment, then the space taken will vary with the size of the field's contents.
5. If the field's contents to be retrieved from the database is of a Wrap or Variable field, the field can be a maximum of 32K characters. (All other fields can be a maximum of 999 characters.)

## Display Format

Specifies a mask for altering the display of date and number values. If the actual value is longer than the specified display format, the value will appear as a string of asterisks in the generated report. The date formats offered in SQL\*ReportWriter are those supported by the TO\_CHAR function in SQL. Number formats are slightly different, but they are more flexible (i.e., you can replace certain characters, etc.).

Default Blank

Options Date formats and format masks are listed below. Capitalization follows the corresponding specified date format. For example, "DAY" in a format model produces capitalized words like "MONDAY"; "Day" produces "Monday," and "day" produces "monday."

Note: The default date and number formats for international languages are changed using the same date and number formats for the default language (U.S.).

The following table illustrates the format masks available for displaying dates

| <b><i>Format Mask</i></b> | <b><i>Meaning</i></b>                                   |
|---------------------------|---------------------------------------------------------|
| SCC or CC                 | Century, abbreviated; 'S' prefixes "BC" with (-)        |
| YYYY to SYYYY             | Year; 'S' prefixes "BC" date with a (-)                 |
| YYY, YY or Y              | Last 3, 2, or 1 digit (s) of year                       |
| Y, YYY                    | Year with comma                                         |
| SYEAR or YEAR             | Year, spelled out; 'S' prefixes "BC" date with (-)      |
| BC, AD, or B.C., A.D.     | Century indicator                                       |
| Q                         | Quarter of year (Jan-Mar=Quarter1)                      |
| M M                       | Month in digits (Jan = 01)                              |
| MONTH or MON              | Name of month, or 3-letter abbreviation                 |
| W W; W                    | Week in year; week in Julian days                       |
| DDD, DD or D              | Day in year, month, or week                             |
| DAY                       | Day of week (MONDAY)                                    |
| DY                        | Name of day, 3-letter abbreviation                      |
| J                         | Julian day;f the number of days since January 1,4712 BC |
| AM, PM, or A. M., P.M.    | Meridian indicator                                      |
| HH or HH12; HH24          | Hour of day (1-12) ; Hour of day (0-23)                 |
| MI                        | Minute                                                  |
| SS; SSSSS                 | Second in minute; seconds in day                        |

The following suffixes may be added to the format models

| <b><i>Suffix</i></b> | <b><i>Explanation</i></b>                                                                                       |
|----------------------|-----------------------------------------------------------------------------------------------------------------|
| fm                   | Toggles fill mode which replaces multiple spaces before or between dates, numbers, or words with a single space |
| TH                   | Suffixed number ("DDth" for "4th")                                                                              |
| SP                   | Spelled outnumber ("DDSP" for "FOUR")                                                                           |
| SPTH or THSP         | Spelled and suffixed number ("DDSPTH" for "FOURTH")                                                             |

The following table illustrates some sample date formats

| <b><i>Sample Date Format</i></b> | <b><i>Sample Display</i></b> |
|----------------------------------|------------------------------|
| MM/DD/YY                         | 03/04/85                     |
| DD MON YYYY                      | 04 MAR 1985                  |
| fm Mon. DD, YYYY                 | Mar. 4, 1985                 |
| fm Day Month DD fmHH: MI AM      | Monday March 4 11:35 AM      |
| fmDy Mon ddth fmHH24 :MI :SS     | Mon Mar 4th 23:35:22         |
| fmDay "thewddthsp "of" Month     | Monday the fourth of March   |

The following table illustrates the display formats for numbers:

| <i>Format</i> | <i>Value</i> | <i>Display As</i> | <i>Explanation</i>                                                              |
|---------------|--------------|-------------------|---------------------------------------------------------------------------------|
| -99           | -34          | -34               | Sign for negative values only                                                   |
| 999           | 34           | 034               | Leading zeros displayed                                                         |
| ZZZ           | 0            |                   | No values displayed                                                             |
| ZZZ9          | 34           | 34                | Leading zeros suppressed                                                        |
| Z,ZZ9         | 1234         | 1,234             | Comma displayed                                                                 |
| +ZZZ9         | 34           | +34               | Sign for positive and negative values                                           |
| \$ZZZ9        | 34           | \$34              | Floating dollar sign                                                            |
| (ZZZ9)        | -34          | (34)              | Negative value in parentheses                                                   |
| ZZZ9.9        | 34           | 34.0              | Decimal point                                                                   |
| ZZZ9.9%       | 34           | 34.0%             | Percent sign                                                                    |
| ZZZ9DB        | 34           | 34DB              | Debit for positive values                                                       |
| ZZZ9CR        | -34          | 34CR              | Credit for negative values                                                      |
| \$BZB9        | 34           | \$ 3 4            | B always adds one space                                                         |
| \$bZZ9        | 34           | \$34              | bZ adds space if there is room                                                  |
| \$FBZZZ9      | 34           | \$ 34             | F keeps \$ fixed at left                                                        |
| \$F-ZZZ9      | 34           | \$-34             | Fixed \$ at left, floating sign                                                 |
| Z"x"x"9       | 34           | 3xx4              | Double-quoted string inserted in value                                          |
| *Z*Z*Z9       | 34           | **34              | *Z replaces one blank with *                                                    |
| 999V99        | 343          | 34300             | Multiplies value times 10 <sup>n</sup> where n is the number of 9's after the V |

## Relative Position

Specifies the position of the field, depending upon the group. If the previous field is in the same group, the position is relative to the previous field. If the field is the first in the group, the position is relative to the left margin of the group.

**Default** Blank implies *Right* if the Print Direction is Down or Down/Across or Below if the Print Direction is Across or Across/Down.

**Options** *Right* places the field on the same line to the right of the previous field, or on the next line to the right of the left margin of the group.

*Below* places the field on the next line at the left margin of the group.

*Panel* places the field on the next panel at the left margin of the group.

**Rules** 1. If *Below* or *Right* is specified and there is no room on the panel or the line, respectively, an error occurs when you generate the report.

2. The meaning of a blank setting is dependent on various settings at the group and/or report level. See the following table.

The following table describes the positioning of fields when the indicated sequence of attributes is set and combined with the Multi-Panel setting.

| <i>Bottom Margin</i> | <i>Right Margin</i> | <i>Fields Across</i> | <i>Multi-panel Blank</i> | <i>x</i> |
|----------------------|---------------------|----------------------|--------------------------|----------|
| Room                 | Room                | Not exe.             | Right                    | Right    |
| Room                 | Room                | Exceeded             | Below                    | Panel    |
| Room                 | No room             | Not exe.             | Below                    | Panel    |
| Room                 | No room             | Exceeded             | Below                    | Panel    |
| No room              | Room                | Not exe.             | Right                    | Right    |
| No room              | Room                | Exceeded             | Error                    | Panel    |
| No room              | No room             | Not exe.             | Error                    | Panel    |
| No room              | No room             | Exceeded             | Error                    | Panel    |

For example, the first line specifies that when there is room left before the bottom and right margins, and the number of fields across the panel has not exceeded the setting, and if Multi-Panel is left blank or selected with an X, then the field will print to the right of the previous field.

The next line specifies that if there is room on both margins but the number of fields across has been exceeded, the field will print below the previous field if Multi-Panel is not selected, and will print on the next panel if Multi-Panel is selected.

**Note:** The preceding table applies to Down and *Down/Across* groups. For *Across* and *Across/Down* groups, replace Right with Below, and replace *Below* with Top and Right. Top and Right means to wrap to the top of the group and to the right of the widest label.

### Lines Before

- Specifies the number of lines before a field.
- Default** Blank implies no lines before the field.
- Options** Any number between zero and the number of lines left on the page, minus the number of lines needed by the field for *Across* and *Across/Down* groups.

### Spaces Before

- Specifies the number of spaces before a field.
- Default** Blank implies the number of spaces set for field spacing within the group.
- Options** Any number between zero and the number of spaces left on the page, minus the number of spaces needed by the field for *Down* and *Down/Across* groups.

## Field Screen Three

Use Field Screen Three (Figure 6-4) to specify the following attributes:

- align fields
- skip fields
- repeat fields on all panels
- compute fields with a function and a reset group.

**FIGURE 6-4**  
Field Screen Three

| Action Query Group Field Summary Text Report Parameter Help                                                            |       |      |        |                 |                |             |
|------------------------------------------------------------------------------------------------------------------------|-------|------|--------|-----------------|----------------|-------------|
| Field Settings                                                                                                         |       |      |        |                 |                |             |
| Field Name                                                                                                             | Align | skip | Repeat | Function        | Computed Value | Reset Group |
| A                                                                                                                      |       |      |        |                 |                |             |
| u                                                                                                                      |       |      |        |                 |                |             |
| <div style="border: 1px solid black; padding: 2px;">           Choose the justification for this field.         </div> |       |      |        |                 |                |             |
| Report Name:                                                                                                           |       |      |        | <List><Replace> |                |             |

### Align

Specifies the justification of the field. Each datatype has its own default value.

**Default** Default is as follows:

- Left for character and date values
- Right for number values.

For matrix reports, the default alignment for the fields of the *Across* group is centered within the width of the cell group.

**Options** Left causes the field's contents to be left-justified within the defined field width.

*Right* causes the field's contents to be right-justified within the defined field width.

Center causes the field's contents to be centered within the defined field width.

Wrap causes a field's contents to be formatted onto multiple lines if necessary. This occurs when there are more characters in the field than the Field Width.

*Variable* causes the contents of the field to be combined with other fields and text into one or more paragraphs without leading blanks. See "Text Object" in Chapter 8 for details.

- Rules**
1. If a Wrap field's data contains soft returns, they will be treated as hard returns.
  2. If a Variable field's data contains &CR's, they will be treated as regular text and not as hard returns.
  3. Variable text fields can contain hard returns and soft returns. Soft returns (i.e., normal blank lines) produce a space unless they are followed by a hard return, in which case the soft return is discarded. On the Text Screen, you embed hard returns into the text object as &CR. Two soft returns in a row cause the same effect as one hard return. In general,  $n$  consecutive soft returns produce  $(n-1)$  hard returns when  $n > 1$ . For example, if there were four soft returns in a row, they would be treated as three hard returns.
  4. Variable fields will cause all text and/or fields in a text object to be word-wrapped into a single paragraph. If you wish to avoid the formation of a single paragraph, you can do one of the following  
1) put a hard return at the end of each line of text that follows the variable-width field; or  
2) create a new group and assign the fields and text that come after the Variable field to the new group. Hard returns are inserted by embedding &cR in text.
  5. *The* contents are wrapped on word boundaries, i.e., on spaces, hyphens, and tabs. Other punctuation marks are considered to be the last character of the word they follow. New line characters in the field's contents force new lines. There is no limit to the number of lines that can be created (other than the actual page height).
  6. Since all fields and text in a text object are wrapped to form one paragraph (if the object contains a Variable field), create additional groups to separate the fields to be wrapped together from those to be kept separate, or embed hard returns.
  7. Wrap and Variable fields are not allowed in Matrix, Across, and Across/Down reports.

8. Wrap and Variable fields in Page Headers and Page Footers will be truncated if the data causes the text to expand beyond the default length for these objects. The default length for Page Headers and Page Footers that contain Wrap or Variable fields is calculated in the following way the height is the number of hard returns plus one. Otherwise, it is the number of soft returns.
9. If a Wrap or Variable field is going to wrap onto the next page, the entire text object to which the field belongs will be moved to the next page, provided the Wrap or Variable field is smaller than the size of a page. If a Wrap or Variable field is greater than the size of a page, it is truncated to fit that page.
10. If a Wrap field appears in the same text object as a Variable field, then it is treated as a Variable field.
11. Text objects containing Wrap or Variable fields are formatted left to right, and then top to bottom. This is true even when several wrapped or variable fields are mixed with text in text objects. The effect is that all text and field values are wrapped to fit within the width of the text. See example one, below.
12. Wrap and Variable fields can contain a maximum of 32K characters.
13. Fields that are placed below a Wrap field in a text object will overwrite the corresponding line of the Wrap field. (See "Examples of the Wrap Setting" on the following page, number 5.)
14. Wrap and Variable fields are not supported in Multi-panel reports.

Examples The following five examples are of Wrap fields in a text object. Field descriptions for the five examples

| <i>Field</i> | <i>Data Type</i> | <i>Width</i> | <i>Align</i> | <i>Contents</i>                     |
|--------------|------------------|--------------|--------------|-------------------------------------|
| ENAME        | CHAR             | 7            | LEFT         | TIMOTHY                             |
| COMMENT      | CHAR             | 14           | WRAP         | AN OUTSTANDING EMPLOYEE . THE BEST. |
| LOC          | CHAR             | 10           | LEFT         | WASHINGTON                          |

Note: The text object containing these fields is defined to be 33 characters wide.

## Examples of the Wrap Setting

| <i>Body</i>                 | <i>Output</i>                                                |
|-----------------------------|--------------------------------------------------------------|
|                             | [----- 33 -----]                                             |
| 1. &ENAME &COMMENT &LOC     | TIMOTHY AN OUTSTANDING WASHINGTON<br>EMPLOYEE. THE<br>BEST . |
| 2. &ENAME &COMMENT<br>& LOC | TIMOTHY AN OUTSTANDING<br>WASHINGTONEMPLOYEE. THE<br>BEST .  |
| 3. &LOC \$COMMENT<br>&ENAME | WASHINGTON AN OUTSTANDING<br>TIMOTHY EMPLOYEE. THE<br>BEST . |
| 4. &ENAME &COMMENT<br>& LOC | TIMOTHY AN OUTSTANDING<br>WASHINGTON EMPLOYEE- THE<br>BEST.  |
| 5. &COMMENT<br>& LOC        | AN OUTSTANDING<br>WASHINGTON THE<br>BEST .                   |

The following two examples are of Variable fields in a text object. Field descriptions for the two examples:

| <i>Field</i> | <i>Data Type</i> | <i>Width</i> | <i>Align</i> | <i>Contents</i>                    |
|--------------|------------------|--------------|--------------|------------------------------------|
| ENAME        | CHAR             | 7            | LEFT         | TIMOTHY                            |
| COMMENT      | CHAR             | 14           | VARIABLE     | AN OUTSTANDING EMPLOYEE. THE BEST. |
| LOC          | CHAR             | 10           | LEFT         | WASHINGTON                         |

Note: The text object containing these fields is defined to be 33 characters wide.

## Examples of the Variable Setting

| <i>Body</i>                                                   | <i>Output</i>                                               |
|---------------------------------------------------------------|-------------------------------------------------------------|
|                                                               | [ <u>                  </u> 33 <u>                  </u> ]  |
| 1. &ENAME &COMMENT &LOC                                       | TIMOTHY AN OUTSTANDING EMPLOYEE<br>THE BEST. WASHINGTON     |
| 2. Did you know that<br>&ENAME is an<br>outstanding employee? | Did you know that TIMOTHY is an<br>outstanding employee?    |
| 3. &ENAME&CR &COMMENT &LOC                                    | TIMOTHY<br>AN OUTSTANDING EMPLOYEE. THE<br>BEST. WASHINGTON |
| 4. &ENAME<br>&COMMENT<br>& LOC                                | TIMOTHY AN OUTSTANDING EMPLOYEE.<br>THE BEST. WASHINGTON    |

### skip

Specifies that SQL\*ReportWriter should not print the indicated field. You skip fields when they are needed for use in computed fields and/or summaries, but you do not want them to appear in the report.

**Default** Blank, implying that the field will redisplayed.

**Options** X skips the field in the report output.

### Repeat

Causes the field to appear on all the panels of the page as a label column. This is useful for simulating a spread sheet style of output.

**Default** Blank, implying that the field will be displayed on the first panel only.

**Options** X prints the field on all panels.

### Computed Value

Computed fields show the results of calculations on values after they are used to perform calculations on query results. These calculations augment the kinds of calculations that can be done directly with a SELECT statement. (User exits, which are another way to augment calculations, are described in Appendix E.) A compute function cannot be specified on a field whose Source is a user exit, or on a field whose Source is &DATE, &PAGE, or &NUM\_PAGES. If you wish to do so, create a Summary on the Summary Screen of the field associated with the user exit field, or the field whose Source is &DATE, &PAGE, or &NUM\_PAGES.

## Function

Specifies the function to use when computing a value of the field. This column is ignored when the Source column contains a reference to a user exit.

**Default** Blank, implying no function is to be computed.

|                |            |                |
|----------------|------------|----------------|
| <b>Options</b> | S u m      | R _ s u m      |
|                | M i n      | R _ m i n      |
|                | M a x      | R _ m a x      |
|                | C o u n t  | R _ c o u n t  |
|                | A v g      | R _ a v g      |
|                | %T o t a l | R _ %t o f a l |
|                | F i r s t  | L a s t        |

The R\_ indicates the cumulative or running version of the function. Absence of the R\_ indicates the periodic version of the function.

The periodic functions allow aggregate results to be printed at a detail level. For example, the DEPT average can be printed on the same line as the EMP detail.

Sum calculates the total of the value in the field within the reset group. R\_sum calculates the running total.

Min calculates the minimum value of the data in each instance of the reset group. R\_min calculates the running minimum of the data in each instance of the reset group.

Max calculates the maximum value of the data within the reset group. R\_max calculates the running maximum of the data within the reset group.

Count counts the number of records within an instance of the reset group. R\_count provides a running count of the data within the reset group; if the reset group is the group you are counting, then R\_Count provides a ranking.

Avg calculates the average of the values within the reset group. R\_avg calculates the running average of the values.

%Total calculates the percent of total for the field within the reset group. For example, each EMP can be computed as a percent of the DEPT group by specifying the function %Total and a reset group of DEPT.

First prints the first value found in each instance of the reset group.

Last prints the last value found in each instance of the reset group.

- Rules**
1. The running functions print the value as it accumulates. R\_max provides the largest “so far” in the list, and so on.
  2. SUM, AVG, and %TOTAL (and their running functions) can only be applied to fields with a datatype of NUM.
  3. You cannot make a computed field on a field associated with a user exit, or on a field whose Source is &DATE, &PAGE, or &NUM\_PAGES. To summarize a user exit field, you should create a summary field on the Summary Screen.

## **Reset Group**

Specifies the group at which to reset the computed field. Report maybe specified to request that the values be calculated for the report as a whole (i.e., for grand totals). *Page may be specified* to request that the values be calculated for each page of the report to achieve page totals.

Default For fields in the highest group, *Report*. For fields in other groups, the next higher group on the same path of the query hierarchy.

Options Any valid group name  
*Report*  
*Page*

- Rules**
1. The reset group must be higher than the lowest group of the query that contains the source column you are summarizing.
  2. The reset group must be higher than the print group for simple summaries.
  3. The reset group can be equal to or higher than the print group for running summaries.

CHAPTER

# 7

## SUMMARY SCREENS

This chapter discusses the Summary Screens. Use these screens to:

- create periodic and running summaries for fields
- create multiple summaries for each field
- specify where each summary prints and when each summary resets to zero.

## Summary Management

A summary field contains data derived by SQL\*ReportWriter using one of the summary functions. Summary fields are not created by default, and if desired, must be inserted and deleted manually. Multiple summaries can be computed for each field.

### Creating a Summary

To create a summary do the following:

1. Use [Insert Record Above] or [Insert Record Below] to make room for a new record; if there are no existing summary fields, skip to the next step.
2. Enter a summary name.
3. Specify the field to summarize.
4. Enter the function used to compute the summary.
5. Enter the field width and display format.
6. Specify the print group. The summary appears by default once for each instance of the print group.
7. Specify the reset group, that is, the group at which the summary resets to zero.

### Deleting a Summary

Use [Next Record] or [Previous Record] to position the cursor at the summary to be deleted, and press [Delete Record] to delete the summary.

### Moving a Summary

To move a summary,

1. Place the cursor to the summary you want to move and press [Delete Record].
2. Position the cursor above the line you want the deleted summary to appear and press [Undelete Record].

### Renaming a Summary

To rename a summary, type the new name over the existing name on Summary Screen One. All text objects that were not edited will be automatically updated to reflect the new name. If you edited a text object that contains the old summary name, you must go to that text object and type the new summary name over the old summary name.

## **Creating Multiple Summaries**

To create multiple summaries from one field, enter multiple records that reference the same field. The different summaries can have the same or different functions, reset groups, and print groups.

## **Positioning Summaries**

Summaries are not created by default. Once you have created a summary field, SQL\*ReportWriter automatically formats the summary as described in the next two paragraphs.

For non-matrix reports, summaries are placed in the group footer of the child of the print group. Summaries with the same function are placed on the same line for Down and Down/Across groups, or in the same column for Across and Across/Down groups. If there is not enough room on the current line to properly format summary in a Down or Down/Across group, a new line is created for that summary. If there is not enough room in the current column to properly format a summary in an Across or Across/Down group, a new column is created for that summary.

For matrix reports, summaries are placed in the group subfooter of the print group rather than the group footer of the child. This default formatting simplifies the process of adding subtotals and grandtotals to reports.

Summaries are placed in text objects just like fields. Override the default positioning by editing the text objects. This includes inserting them in other than the default texts. You can refer to a summary in a text object by preceding the summary name with an ampersand (&).

---

## **Summary Screen One**

Summary Screen One (Figure 7-1) is used to specify the following attributes

- summary name
- field to summarize
- function to use when summarizing
- datatype of the summary field
- width and display format for the summary field.

**FIGURE 7-1**  
**Summary Screen One**

| Summary Settings |       |          |           |       |                |
|------------------|-------|----------|-----------|-------|----------------|
| Summary Name     | Field | Function | Date Type | Width | Display Format |
|                  |       |          |           |       |                |

Enter a name for this summary field.  
Report Name:  <Replace>

**Summary Name**

Specifies the name of the summary. Summaries are named so they can be referenced in other parts of the report definition.

**Default** None. No summaries are created by default.

**Options** A List of values is provided showing the names of existing summaries. Do not use these names for a new summary.

**Rules** Summary names must be distinct from the report name, query names, group names, field names, and other summary names.

**Field**

Specifies the name of the field being summarized. Summaries are based on fields, not on expressions or user exits.

**Default** None.

**Options** Any field that appears on the Field Settings Screen. (Choose a field from the List of values, which shows a list of all existing fields.)

- Rules**
1. The field you enter must be an existing field.
  2. If the field that is being summarized has a Wrap or Variable alignment, the summary will have Wrap or Variable alignment, respectively.
  3. The field name you enter must not have a Source of &DATE, &PAGE, or &NUM\_PAGES. (The Source setting is found on Field screen One.)

## Function

**Specifies the function used to compute the summary.**

Default

Blank

Options

|               |                  |
|---------------|------------------|
| <i>Sum</i>    | <i>R_sum</i>     |
| <i>Min</i>    | <i>R_min</i>     |
| <i>Max</i>    | <i>R_max</i>     |
| <i>Count</i>  | <i>R_count</i>   |
| <i>Avg</i>    | <i>R_avg</i>     |
| <i>%Total</i> | <i>R_% Total</i> |
| <i>First</i>  | <i>Last</i>      |

The R\_ indicates a running summary. The running functions print the value as it accumulates.

The periodic functions allow aggregate results to be printed at a detail level.

Sum calculates the total of the values within each instance of the reset group. R\_sum calculates the running total.

Min calculates the minimum of the values within each instance of the reset group. R\_min calculates the running minimum.

Max calculates the maximum of the values within each instance of the reset group. R\_max calculates the running maximum.

Count calculates the number of values within each instance of the reset group. R\_count calculates the running count.

Avg calculates the average of the values within each instance of the reset group. R\_avg calculates the running average.

%Total calculates the percent of total for each value within each instance of the reset group. R\_% Total computes the running percent of the total.

First causes the value from the first record of the reset group to be printed.

Last causes the value from the last record of the reset group to be printed.

Rules

1. The running functions print the value as it accumulates. R\_Max provides the largest "so far" in the list, and so on.
2. SUM, AVG, %TOTAL, and all running functions can only be applied to columns with a datatype of NUM.

|                       |                                                                                                                                                                                                                                                                     |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Data Type</b>      | Specifies the data type for the summary field.                                                                                                                                                                                                                      |
| <b>Default</b>        | Same as the datatype of the field being summarized.                                                                                                                                                                                                                 |
| <b>Options</b>        | None. The values are maintained by SQL*ReportWriter.                                                                                                                                                                                                                |
| <b>Rules</b>          | <ol style="list-style-type: none"> <li>1. SQL*ReportWriter selects the datatype for the summary based on the datatype of the field being summarized.</li> <li>2. If the field being summarized is changed, the datatype of the summary changes to match.</li> </ol> |
| <b>Width</b>          | The number of characters the summary will occupy.                                                                                                                                                                                                                   |
| <b>Default</b>        | Same as the width of the field being summarized.                                                                                                                                                                                                                    |
| <b>Rules</b>          | If the value to be displayed is greater than the width of the summary, SQL*ReportWriter truncates characters and dates or replaces the value with asterisks if the datatype is NUM.                                                                                 |
| <b>Display Format</b> | Specifies a mask for the display of date and number values. Date formats offered in SQL*ReportWriter are a superset of those in the TO_CHAR function in SQL. Number formats are slightly different.                                                                 |
| <b>Default</b>        | None.                                                                                                                                                                                                                                                               |
| <b>Options</b>        | Refer to Display Format in Chapter 6, "Field Screens;" for a complete listing of date and number formats.                                                                                                                                                           |

## Summary Screen Two

Summary Screen Two (Figure 7-2) is used to specify the group(s) at which SQL\*ReportWriter resets the summary to zero and prints the summary.

**FIGURE 7-2**  
Summary Screen Two

| Action                                                       | Query | Group        | Field       | Summary     | Text | Report | Parameter       | Help |
|--------------------------------------------------------------|-------|--------------|-------------|-------------|------|--------|-----------------|------|
| <b>Summary Settings</b>                                      |       |              |             |             |      |        |                 |      |
|                                                              |       | Summary Name | Print Group | Reset Group |      |        |                 |      |
| Enter the name of the group where this summary is displayed. |       |              |             |             |      |        |                 |      |
| Report Name:                                                 |       |              |             |             |      |        | <List><Replace> |      |

## Print Group

The print group determines how often the summary will appear in the report. The summary will print once for each record in the print group.

**Default** The parent of the group containing the field being summarized, or *Report*. For running summaries of fields within the top group, it is the top group.

**Options** A List of values is provided showing the names of all the valid groups, as well as *Report*.

*Report* prints the summary once at the end of the report.

**Note:** The Print Group must be above the group containing the field being summarized in the same path, or *Report*.

- Rules**
1. The print group must not be above the reset group in the group list.
  2. If the print group is the same as the reset group, the value is reset each time it is printed.
  3. For non-matrix reports, summaries are placed in the group footer of the immediate child of the print group.

4. In matrix reports, summaries whose Print Group has a Print Direction of Down or Across appear as column totals or row totals, respectively, and are placed in the subfoot of the Print Group itself.
5. In matrix reports, summaries whose Print Group is Report are placed in the footer of the group whose Print Direction is Crosstab.
6. Page summary references must be manually inserted into a text object.
7. When the Print Group is the same as the Reset Group, the value is accumulated until a new value of the group is found. It is then printed, reset to zero, and the accumulation begins again.
8. When the Reset Group is above the Print Group, running and periodic functions behave differently. For running functions, the value is printed each time a new value of the Print Group is found. For periodic functions, a forward reference occurs. SQL\*ReportWriter waits until a new record in the Reset Group occurs, and then copies the fully accumulated value back into each record of the Print Group.

**Reset Group**

Specifies the name of the group at which the summary resets to zero. Specify the keyword *Report* to request grand total values for the report as a whole. Specify the keyword *Page* to request page totals.

**Default**

For fields in the highest group, *Report*. For other fields, the next higher group in the same path. For running functions, the grandparent of the group containing the field being summarized.

**Options**

A List of values is provided showing the names of all valid groups in addition to *Report* and *Page*.

A group name resets the summary at the specified group.

*Report* resets the summary at the end of the report.

*Page* resets the summary at the end of the page.

**Rules**

1. The reset group must be a valid group.
2. The group of the field being summarized must be in the same path in the group list as the reset group.
3. You must enter a Print Group before you can enter a Reset Group.

# 8

## TEXT SCREEN

The Text Screen allows you to customize your report by editing default text objects and creating your own text. You use this screen to do the following

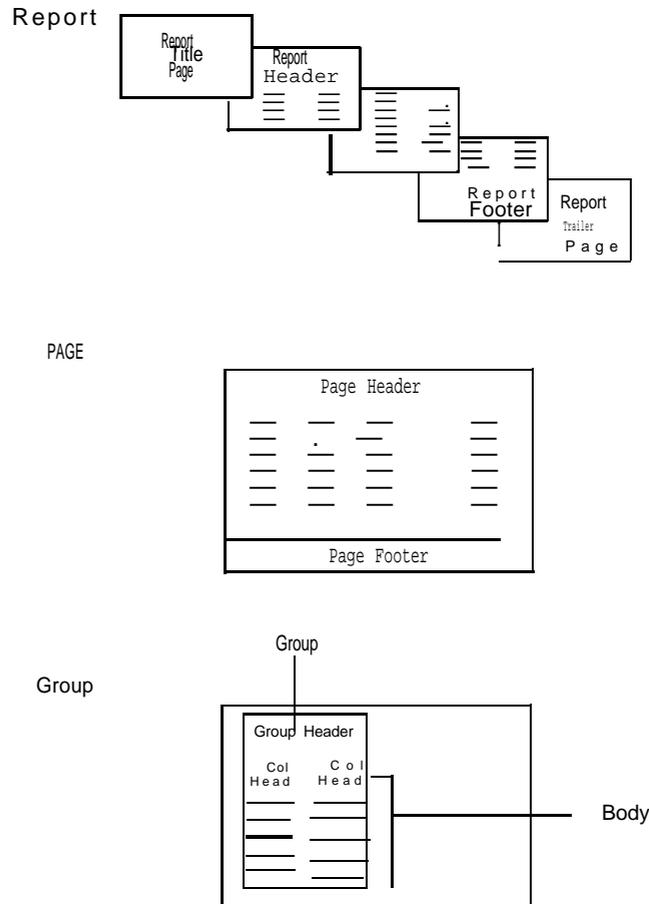
- modify text objects
- modify text and field locations in text panels
- create, delete, and edit text *panels*
- assign highlight attributes to fields and text.

## Text Objects

The following is the list of text objects

- **Report Title Page** occurs at the beginning of a report.
- **Report Header** occurs at the beginning of a page, report, or group.
- **Column heading** concatenates the field labels for the fields in a group.
- **Body** occurs once for each record of a group.
- **Subfooters** appear only in matrix reports for the Across group and the Down group. Subfooters print after all the rows or columns of the group.
- **Report Footer** occurs at the end of a page, report, or group.
- **Trailer Page** occurs at the end of a report.

**FIGURE 8-1**  
Text Objects





## **Text Management**

SQL\*ReportWriter creates text for many objects by default, including Column Headings (if the default Label Position is used) and a Body for every group, and Group Footers for summaries. Text for other objects must be created manually.

Some reports are wider than a single page. SQL\*ReportWriter creates one panel for each of the overflow pages for default objects. For other objects, panels must be created manually.

## **Locating Text Objects**

In order to locate a particular text object, press [Next Record] or [Previous Record] repeatedly until the object appears. Alternatively, press [Query] to clear the Object and Type fields and to place SQL\*ReportWriter into query mode. Enter the name and/or the Type of the object you are trying to locate, and press [Fetch]. SQL\*ReportWriter will now fetch only a subset of all text objects based on the search criteria you entered. To return to the full set, press [Query] [Fetch].

## **Suppressing Printing of a Default Panel**

To suppress the printing of a default object, delete all of the characters in all of the panels associated with the object without deleting the panels themselves. You can identify such blank panels because the panel will be numbered and the Status will be "Edited," even though no text is shown.

## **Customizing Your Report**

To customize a text object, use [Next Record] or [previous Record] to locate the panel to be changed and then revise the text. Use the Text editing keys ([Delete Line], etc.) to revise text. Pressing [Delete Record] will cause the entire text object to be restored to its default (i.e., un-edited) state when the report is executed, or when the screen is recentered.

## Creating Text Panels

SQL\*ReportWriter computes the number of panels in a report based on various group and field settings, and creates the proper number of panels for each text object that has a default value. For other objects, such as Page Headers, you must insert each panel explicitly.

To create panels for page headers, footers, and other objects not created automatically by SQL\*ReportWriter, do the following

1. Locate the desired object, for example, Report Footer, using [Next Record] and [Previous Record] (or [Query] and [Fetch]).
2. Use [Insert Record Below] to create a panel to hold the text and enter the appropriate panel number.
3. Enter the text.
4. Repeat these steps for each panel needed.

## Re-creating Default Text

To delete the edited objects of a panel and restore the default behavior, locate each edited panel of the text object and use [Delete Record]. You must explicitly delete all edited panels of a text object in order to regenerate default versions of that text object. SQL\*ReportWriter recreates default text objects automatically. Panels for text objects that have no default (that were created manually such as a Page Header) are not re-created if they are deleted. Do the following to restore default text objects:

1. Delete all the edited panels belonging to the edited text object, by pressing [Delete Record].
2. Commit your changes to the Text Screen by pressing [Accept] or by pressing [Menu] and then entering another screen.
3. Return to the Text Screen; the text object will be in its default state.

Now, when you execute the report, all changes will be included in the output, as well as the edited text.

For best results, try to define all fields, groups, and summaries before you edit any text objects. If you do make subsequent changes, such as inserting or removing fields, make the changes manually in the text object before you execute the report. For example, if you have already edited a text object and you then decide you would like a computed field to be printed in the same body text, it is simpler to insert the new field reference into the text and edit the appropriate column heading than to delete all of the text and then re-edit the new defaults.

## **Locating Specific Text Objects**

If you want to retrieve a specific text object, do the following:

1. Press [Query].
2. Use the List of values to select the object and/or text type you wish to retrieve. You can also enter values for the other settings, except Status and the text itself. Wildcards are not supported.
3. Press [Fetch].

## **Highlighting Text and Fields**

There are four ways of highlighting text and data in a report: 1) select a highlight from the Highlight Text List of values; 2) embed printer control codes (e.g., &O1, &O2, etc.) in the text; 3) pass printer control codes from a user exit; and 4) enter highlighting options for Group field labels or fields. The first two ways to highlight text and data can be done using the Text Screen. For more information on printer control codes from user exits, see "Conditionally Highlighting Fields" in Appendix E. For more information on entering highlighting options for Group labels or fields, see "Highlight" in Chapter 5.

The first option, selecting a highlight from the Highlight Text List of values, enables you to highlight your text and database fields in eight different ways, and to see the effect of your changes on your screen.

Printer control codes, the second option mentioned, allow more flexibility but are somewhat less convenient to use. This is because the effect of a printer code is not seen unless the report is printed using a printer definition that defines an escape sequence for that code. If your report is viewed in the browser or if the codes are not defined in the printer definition, they are ignored.

Each SQL\*ReportWriter administrator can define as many printer control codes as desired, assigning any escape sequence or series of control characters to each code. Thus, by embedding these printer codes in your report, you will be able to take full advantage of the capabilities of your printer such as font sizes and styles.

To highlight text and data by using the Highlight Text List of values, do the following:

## **Using Highlight Text List of Values**

1. Place the cursor on the first character of the text you wish to highlight and press [Mark]. Text will appear in reverse-video from the first character to be highlighted to the current cursor position.
2. Use [Right], [Left], [Next Line], or [Previous Line] to move the cursor to the the last character of the text you wish to highlight and press [Highlight].

The Highlight Text List of Values appears. Your highlighting options are

|                      |                               |
|----------------------|-------------------------------|
| <i>Current Style</i> | <i>Underline/Reverse</i>      |
| <i>Normal</i>        | <i>Underline/Bold</i>         |
| <i>Underline</i>     | <i>Reverse/Bold</i>           |
| <i>Bold</i>          | <i>Underline/Reverse/Bold</i> |

3. Use [Next Choice] or [Previous Choice] to move the cursor to the highlighting style of your choice and then press [Select].

Note Current Style is the one you selected prior to this operation, and is indicated on the status line.

## **Embedding Printer Control Codes**

To highlight text and data by embedding printer control codes in the text, do the following:

Place the cursor before the text or database field you wish to highlight and enter a printer control code. You must insert a non-numeric character after the printer code. All printer codes are named "&nnn..." where "nnn..." is any number. Examples of printer codes are &1, &002, &03, &3, etc. There is no limit to the number of printer codes you can define and use.

Printer codes are customized for each SQL\*ReportWriter installation, so ask your System Administrator for a list of codes and the effect of each when printed.

- Rules**
1. Zeros immediately to the right of the ampersand are ignored, thus, &1 and &01 are equivalent.
  2. The text that follows the printer code will continue to be highlighted until another printer control code changes that highlight.
  3. If a printer code has not been defined by your System Administrator, SQL\*ReportWriter will ignore the code and discard the reference from the report output when executing your report. The code will remain, however, in the text object.
  4. Printer codes can be thought of as fields with a width of zero. Thus, their width is ignored by SQL\*ReportWriter when it computes or checks the widths of text objects.

## The Text Screen

The Text Screen is divided into two parts: the top section of the screen describes the different text objects that *make* up a report, and the bottom section of the screen contains each of the panels that makeup the text. Each panel is defined by a panel number and its associated text. Use the Text Screen (Figure 8-2) to define text objects, and to insert text and modify default text objects.

**FIGURE 8-2**  
**Text Screen**

| Action                                                            | Query | Group | Field                     | Summary | Text | Report            | Parameter | Help |
|-------------------------------------------------------------------|-------|-------|---------------------------|---------|------|-------------------|-----------|------|
| <b>Text Settings</b>                                              |       |       |                           |         |      |                   |           |      |
| Object: PAGE                                                      |       |       | Type: Header              |         |      | Status: Default   |           |      |
| Relative Position:                                                |       |       | Repeate on Page Overflow: |         |      |                   |           |      |
| Lines Before:                                                     |       |       | Justification: Left       |         |      |                   |           |      |
| Spaces Before:                                                    |       |       | Frequency:                |         |      |                   |           |      |
| Width:                                                            |       |       |                           |         |      |                   |           |      |
| Panel Number:                                                     |       |       | Text                      |         |      | Panels Defined: 0 |           |      |
| Choose the position of the text in relation to the previous text. |       |       |                           |         |      |                   |           |      |
| Report Name:                                                      |       |       | < N o r m a l >           |         |      | <List><Replace>   |           |      |

## Object

Specifies the name of the object that owns the text.

**Default** None. This field can be entered only when in query mode.

**Options** A List of values is provided (in Query mode) showing Report, *Page*, and all valid group names.

*Report* indicates that the text object will be associated with the report as a whole.

*Page* indicates that the text object will be associated with each page of the report.

A group name indicates the text object will be associated with the specified group.

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Text Type</b> | The type of text associated with report, page, or group objects                                                                                                                                                                                                                                                                                                                                                   |
| <b>Default</b>   | None. This field can only be entered in query mode.                                                                                                                                                                                                                                                                                                                                                               |
| <b>Options</b>   | <p>A List of values is provided showing valid text types for the current object. The following text types are available for the associated object types:</p> <p>For the report : Title Page<br/>Header<br/>Footer<br/>Trailer Page</p> <p>For the page: Header<br/>Footer</p> <p>For groups: Header<br/>Column Heading<br/>Body<br/>Subfoot<br/>Footer</p>                                                        |
| <b>Rules</b>     | The specified text type must be valid for the object. Use the List of values to see the valid text type for a particular object.                                                                                                                                                                                                                                                                                  |
| <b>Status</b>    | Indicates whether the default text has been edited.                                                                                                                                                                                                                                                                                                                                                               |
| <b>Default</b>   | Default.                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Options</b>   | This field is non-enterable. When you edit any text of a particular text object and type, the status setting of that particular text object and type changes from Default to Edited.                                                                                                                                                                                                                              |
| <b>Rules</b>     | If the current text object has been edited and you make subsequent changes to formatting attributes of fields, groups, or summaries, those changes will not be reflected in that already edited text object unless you follow the procedure described earlier in this chapter under "Recreating Default Text." All text objects that have not been edited will continue to use the default formatting attributes. |

## Relative Position

Specifies the location of the text object relative to the object that occurs before it.

**Default** Blank, which implies Below for all non-group objects. See Rule 2 for placement of group objects.

**Options** Right places the text to the right of the text preceding it.

Below places the text below the text preceding it.

Page places the text essentially on a new page, separate from all prior data. Text objects can be divided across multiple panels. If Page is selected, the first chunk of the text object will be placed on a new page.

Margin causes the text to appear at the left margin of the page for groups with a Print Direction of Down or Down/Across. Margin causes the text to appear at the top margin of the page for groups with a Print Direction of Across or Accross/Dounz.

- Rules**
1. Not all Relative Positions are valid for all text objects. A List of values is provided for each text object.
  2. The default positioning of Group texts depends on the group's print direction. If the print direction is Down, then the relative positioning is Below; if Across, then Right.
  3. If the Relative Position is set to *Margin*, text may overwrite text of a prior sibling or parent. This enables you to place a text object to the left of a parent or a sibling. To avoid overwriting specify the Lines Before for the Margin text object to be equal to the number of lines of the overwritten text object.

## Lines Before

Specifies the number of blank lines before the text.

**Default** 0

**Options** Any number between 1 and the number of lines remaining on the page, minus the number of lines in the text.

- Rules**
1. In a 'Master-Detail report, if a parent is not printed on a page, its text is compressed to take up no room, but its Lines Before and Spaces Before are still formatted. To avoid this "inheritance," add carriage returns and spaces to the desired text object to achieve the same effect as the Lines Before and Spaces Before settings.

**Spaces Before**

Specifies the number of spaces before the text.

**Default** 0

**Options** Any number between 1 and the number of spaces remaining on the page, minus the number of characters in the text.

**Rules** 1. In a Master-Detail report, if a parent is not printed on a page, its text is compressed to take up no room, but its Lines Before and Spaces Before are still formatted. To avoid this "inheritance:" add carriage returns and spaces to the desired text object to achieve the same effect as the Lines Before and Spaces Before settings.

**Width**

Specifies the width of the text object.

**Default** Width of the text object. The default width of a text object without Variable fields is the width of the widest line in that object. The line widths are calculated by adding the length of boilerplate text of the line with the sum of the widths of the fields on the line. For example, if the longest line was "abc &TOTAL" and the field TOTAL had a Field Width of five characters, the default width would be nine (three for the text, one for the space between the text and field, five for the field width).

The default width for text with Variable fields is the widest line, calculated in the following way: adding the length of boilerplate text of a line, plus the sum of the widths of any fixed-width fields on that line, plus the Field Width of the Variable field.

**Options** Any number between 1 and the width of the report as specified in the Report Settings Screen.

**Rules** 1. This field is optional; if nothing is entered, SQL\*ReportWriter will use its own algorithms to compute the width of the object. If a number is entered, SQL\*ReportWriter will use that text Width for all panels of that text object.

2. If multiple groups are placed side-by-side in your report, and you specify widths for each of them, the combined width of these groups must not exceed the page Width in the Report Setting Screen or you will get an error.

3. If there are no Variable fields in the text object, the text object (including fields) will be truncated at the Width you specify.

## Repeat on Page Overflow

Specifies whether the text should be repeated on the next page if the data overflows.

**Default** X appears for column heading and body; a blank appears for all other text types.

**Options** Blank prints text only on the first occurrence of a group.  
X prints text on overflow pages when there are more records than will fit on the first page.

**Rules** 1. The column heading of a text object inherits the properties of the column heading of the group whose frequency matches.

## Justification

Specifies horizontal alignment for the text.

**Default** Left.

**options** Left  
center  
Right

**Rules** 1. Each line of a text is justified independently within the available region.  
2. For group texts justification is only performed on Down or Down/Across groups; the available region is the width of the group, except for the right-most group on the page. In this case, the available region is the remainder of the page width.

In addition, Justification for group texts is only performed when all four texts (group header, column heading, body, group footer) are positioned on the same page and below one another. Therefore, the Column Heading, Body, and Group Footer must have their Relative Position set to Below.

3. For other texts, the available region is the space between the left and right margins of the page.

**Note:** Because Justification of Center or Right is disabled for matrix and across groups, you must use the alignment setting on the Field Screen if you want the across fields right justified.

## Frequency

Specifies how often the text will be printed. This option is provided to allow column headers to print once per instance of a group, or less frequently.

**Default** Blank, implying once per each instance of a group.

**Options** Report prints the text once above the first occurrence of the current group, except when Repeat on Page Overflow is selected. In this case the text will appear at the top of each page.

Group name prints the text with every record of the selected group.

**Rules** This attribute is only available for column headers.

## Panel Number

Specifies the panel on which to print the text. For Title Page and Trailer Page, specifies the page on which to print the text.

**Default** Default panels are numbered by SQL\*ReportWriter.

**Options**

1. Enter the panel number of manually created panels. The numbers should range from 1 to  $n$  (with a maximum of  $n=10$ ) where  $n$  is the number of panels needed for the report.
2. To resequence the panels, simply type over the current number.

**Rules**

1. The *panel* number is absolute. For example if you have two panels for the same object, one is number 1 and the other is number 4, two blank panels will appear between panel number 1 and panel number 4.
2. Duplicate panel numbers are not allowed.

## Text

The text that is to appear in the report.

**Default** Column headers, bodies, and group footers have defaults which depend on the settings of the group, field, and summary screens. All other text has no default.

**Options** Enter the text to appear as the header, footer, title page text, trailer page text, body, or column header in this area.

Reference any of the following system variables:

- &PAGE references the current page number
- &NUM\_PAGES references the total number of pages
- &DATE references the current date using the default format DD-MON-YY.

The default width and alignment for &PAGE and &NUM\_PAGES is 5 characters, left-justified. The default width and alignment for &DATE is 9 characters, left-justified. You can 'change these defaults by creating a field whose Source is the system variable and then define a different Field Width and/or Display Format.

Reference fields, summaries, and parameters by prefixing the object name with &.

Reference printer codes by prefixing the printer code number with an &. The width of a printer code is O; the height is 1.

Reference fields of Data Type PRT by prefixing the field name with &.

- Rules**
1. Fields and summaries can be referenced either in the Body of their print group or in the Header, Footer, or Body of a group that is below their print group. Fields and summaries can also be referenced in the Column Heading of a group below their print group, if the Column Heading has a Frequency at or below their print group.
  2. You cannot reference fields from a lower group in a higher group; you can only reference summaries of fields from lower groups.
  3. Summaries with a Reset Group of Report or *Page* can be referenced in the Report Header and Footer, Page Header and Footer, Title Page and Trailer Page, in addition to the Footer of the top group (which must be in the same path as the field being summarized).
  4. To reference fields in Page Headers and Page Footers, you must 1) define a new field (on the Field Screen or the Summary Screen) that gets its value from the field to be referenced, 2) specify the First or Last Function for the newly defined field, 3) specify a reset group of PAGE (on the Field Settings Screen), and 4) reference the computed field in the page header or footer. Note: First is the first value of that field to appear on that page. last is the last value of that field to appear on that page.
  5. If a Page Header or Footer contains Variable fields, the header or footer will have a default height of 1 plus the number of hard returns (&CR) in the text object. For all text objects, other than Page Headers or Footers, that contain Variable fields, the text object will have a default minimum height of 1 plus the number of hard returns.

Text objects containing Wrap or Variable fields are formatted left to right, and then top to bottom. The effect is that all text and field values are wrapped to fit within the width of the group. See example one, below.

7. Wrap and Variable fields are not supported in multi-panel reports.
8. If a Wrap field appears in the same text object as a Variable field, then it is treated as a Variable field.
9. Only text objects that contain one or more Variable fields can contain hard returns (A hard return is indicated by &CR or by two soft returns.) This functionality allows you to force blank lines and paragraph breaks as needed. For example, you can have a fixed-width paragraph and a variable-width paragraph in the same report, even in the same group. A "paragraph" of text ends with a hard return and will be formatted by the Align definition(s) of the field(s) that are embedded in the text. See examples of Variable and Wrap fields in text, following.
10. The text field can contain up to 16K characters, new lines and spaces counted.
11. The system variables, &PAGE, &NUM\_PAGES, or &DATE can be referenced in any text object.
12. If you highlight text with a printer control code, the effect of that code will not change until another printer code modifies it or turns it off.

Examples Field descriptions for the next three examples:

| <i>Field</i> | <i>Data Type</i> | <i>Width</i> | <i>Contents</i>                    |
|--------------|------------------|--------------|------------------------------------|
| ENAME        | CHAR             | 12           | TIMOTHY JONES III                  |
| COMMENT      | CHAR             | 14           | AN OUTSTANDING EMPLOYEE. THE BEST. |
| LOC          | CHAR             | 10           | WASHINGTON                         |

**Field Alignment of the following examples:**

1. ENAME is Left, COMMENT is Wrap, LOC is Left.
2. ENAME is Left.
3. ENAME is Variable.

**Note:** The text object containing these fields is defined to be 33 characters wide.

Examples of the Wrap and Variable Align Fields in Text

| <i>Body</i>                                                   | <i>output</i>                                                      |
|---------------------------------------------------------------|--------------------------------------------------------------------|
|                                                               | [-----33-----]                                                     |
| 1. &ENAME &COMMENT &LOC                                       | TIMOTHY JONE AN OUTSTANDING WASHI<br>EMPLOYEE. THE<br>BEST .       |
| 2. Did you know that<br>&ENAME is an<br>outstanding employee? | Did you know that<br>TIMOTHY JONE is an<br>outstanding employee?   |
| 3. Did you know that<br>&ENAME is an<br>outstanding employee? | Did you know that TIMOTHY JONES<br>III is an outstanding employee? |

# 9

## REPORT SCREEN

This chapter discusses the Report Screen. The Report Screen is used to:

- set page dimensions
- specify default margins
- change the title, hint line, and status line of the Run-time Parameter Form
- enter comments to document a report
- grant report access to other users
- display report history.

## The Report Screen

The Report Screen (Figure 9-1) includes the following settings:

- page height and width
- top, bottom, left, and right margins
- parameter title, hint, and status
- comments
- created and modified report history (user, date, version).

**FIGURE 9-1**  
Report Screen

| Action                                 | Query | Group     | Field           | Summary | Text           | Report | Parameter | Help |
|----------------------------------------|-------|-----------|-----------------|---------|----------------|--------|-----------|------|
| <b>Report Settings</b>                 |       |           |                 |         |                |        |           |      |
| Page                                   |       | Marginst  |                 |         | Parameter Form |        |           |      |
| Height: 66                             |       | Top: 2    |                 |         | Title:         |        |           |      |
| Width: 80                              |       | Bottom: 2 |                 |         | Hint:          |        |           |      |
|                                        |       | Left: 0   |                 |         | Status:        |        |           |      |
|                                        |       | Right: 0  |                 |         |                |        |           |      |
| Comments                               |       |           |                 |         |                |        |           |      |
| ^                                      |       |           |                 |         |                |        |           |      |
| v                                      |       |           |                 |         |                |        |           |      |
| Access List                            |       |           | By              |         | History        |        | Ver.      |      |
| ^                                      |       |           |                 | Date    |                |        |           |      |
|                                        |       |           | Created: Scott  |         | 12-Dec-1988    |        | 1.0.03    |      |
|                                        |       |           | Modified: Scott |         | 14-Dec-1988    |        | 1.101     |      |
| Enter the height of the page in lines. |       |           |                 |         |                |        |           |      |
| Report Name:                           |       |           |                 |         |                |        | <Replace> |      |

### Page Height

Specifies the height of the physical page in lines. The page size defines the dimensions of the paper on which the report will be printed.

Default 66

options Enter a value from 1 through 999.

Rules On personal computers running MS-DOS, the product of the page height and page width cannot exceed 32K.

### Page Width

Sets the width of the physical page in character spaces.

Default 80

Options Enter a value from 1 through 999.

Rules On personal computers running MS-DOS, the product of the page height and page width cannot exceed 32K.

**Top Margin** Specifies the number of lines that SQL\*ReportWriter skips from the top of every page before displaying any text or data, including headings.

**Default** 2

**Options** Enter a value from 0 through 999.

**Rules** The sum of the top and *bottom* margins must be less than the page height.

**Bottom Margin** Specifies the number of lines that SQL\*ReportWriter leaves blank at the bottom of every page.

**Default** 2

**Options** Enter a value from 0 through 999.

**Rules** The sum of the top and bottom margins must be less than the page height.

**Left Margin** Specifies the number of spaces that SQL\*ReportWriter skips from the left of every page before displaying data.

**Default** 0

**Options** Enter a value from 0 through 999.

**Rules** The sum of the left and right margins must be less than the page width.

**Right Margin** Specifies the number of spaces that SQL\*ReportWriter leaves blank at the right of every page.

**Default** 0

**Options** Enter a value from 0 through 999.

**Rules** The sum of the left and right margins must be less than the page width.

**Parameter Form Title** Specifies the title of the Run-time Parameter Form. The title appears at the top of the Form in reverse video.

**Default** Blank, implying "Parameter Values"

**Options** Enter any title that is at most 80 characters long (including spaces).  
You can restore the default title if you delete the text you entered.

**Parameter Form Hint** Specifies the hint line of the Run-time Parameter Form. The hint line is displayed on the second to last line of your screen.

**Default** Blank, implying "Enter the desired value for each parameter."

**Options** Enter any text that is at most 80 characters long (including spaces).  
You can restore the default hint if you delete the text you entered.

**Parameter Form Status** Specifies the status line information of the Run-time Parameter Form. The status line appears directly below the hint line, on the last line of your screen.

**Default** Blank, implying "Report Name: <report name>"

**Options** Enter any text that is at most 50 characters long (including spaces).

**Rules** When you change the Parameter Status, the Report Name will not appear in the status line unless you enter it there, yourself. For example, you could enter

Report is printed in room 219B. (Report: deptinfo)

You can restore the default status if you delete the text you entered.

**Comments** Enter text to describe the report in this area. This text is strictly to assist you, and other users, to remember the purpose of your report. The comments do not appear in the report output. This area will scroll vertically and horizontally.

**Default** Blank

**Options** Any text that is less than 32K characters long. Newlines and spaces are counted.

**Access** If you are using system-owned SQL\*ReportWriter tables, you are able to enter an ORACLE username here, to allow that user to have access to your report. To revoke access privileges, simply delete the user's name. The Access field is a list so that you can insert, delete, and scroll back and forth among the names.

If you are using user-owned SQL\*ReportWriter tables, you must use LOADREP or DUMPREP to share reports (see Chapter 4): any entries to this field will be ignored.

**Default** Blank, implying no other users have access to the report.

**Options** Public grants all users access to your report who use SQL\*ReportWriter and share the database with you.

% grants access to all users (just like *Public*). We suggest, however, that you use *Public* for that purpose. Note: An underscore ( `_` ) is equivalent in functionality to a percent sign.

You may also use the wildcard, %, to represent missing characters. For example, if you enter J%, people such as JSIMON and JHARRIS will have access to your report.

Rules If you are granted access to another user's report, you may work with it as you would with your own reports, except that you cannot:

- rename the report
- delete the report
- grant access privileges to that report.

If you copy another user's report (which is only possible if you have access privileges to it), the access privileges are not copied. Therefore, the Access field will be blank, and no-one will have access to that report unless you grant them access. See Chapter 3, "Copy," to learn how to reference other users' reports.

## History

|               |                                                                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------|
| Created       | Displays the ORACLE username of the person who owns the report. This field is non-enterable.                                 |
| Created Date  | Displays the date that the report was created. This field is non-enterable.                                                  |
| Created Ver.  | Displays the version of SQL*ReportWriter that was used to create the report. This field is non-enterable.                    |
| Modified      | Displays the username of the last person who modified the report. This field is non-enterable.                               |
| Modified Date | Displays the date that the report was last modified. This field is non-enterable.                                            |
| Modified Ver. | Displays the version of SQL*ReportWriter that was being used when the report was last modified. This field is non-enterable. |

CHAPTER

# 10

## PARAMETER SCREEN

This chapter first discusses how to manage bind and lexical parameters, and then it discusses the Parameter Screen. Parameters are used to be referenced in queries and text objects, and they are also to provide values at run-time. You use the Parameter Screen to

- view default values of parameters
- specify parameter widths
- alter parameter datatypes
- alter the Run-time Parameter Form
- specify the input and output parameter options.

---

## Parameter Management

Parameters, like other report objects, can be inserted, deleted, renamed, and moved. You can reference parameters in queries and text objects, allowing you to change selection criteria and calculations, for example, at run-time. Parameters may be used to replace expressions in WHERE, GROUP BY, ORDER BY, HAVING, CONNECT BY, and START WITH clauses of queries. There are two types of parameters in SQL\*ReportWriter: bind and lexical.

Bind parameters and lexical parameters are handled differently by SQL\*ReportWriter. With bind parameters, one value is substituted into the parameter reference; with lexical parameters several values may be substituted into the parameter reference. Thus, bind parameters may be used anywhere in the query where a single literal value, such as a character string, number, or date could be used. Lexical parameters can be used in the WHERE, GROUP BY, ORDER BY, HAVING, CONNECT BY, and START WITH clauses, and may replace values as well as SQL expressions. Note that default bind and lexical values may be changed at run-time on the Run-time Parameter Form.

### Bind Parameters

Bind parameter values are substituted each time a new record is fetched from the database (while executing the report). Thus, bind parameter values are changed throughout the execution of a report.

### Lexical Parameters

Lexical parameter values are substituted only once, before the query is parsed. Thus, lexical parameter values are not changed throughout the execution of a report.

### Creating Bind Parameters

To create a bind parameter, you must do the following:

1. Select Query from the Main Menu.
2. Type in a SELECT Statement which uses a parameter, and precede the parameter with a colon.

Bind parameters may be used to replace expressions in SELECT, WHERE, GROUP BY, ORDER BY, HAVING, CONNECT BY, and START WITH clauses of queries. Examples of bind parameters are found below:

**SELECT Clause**

```
SELECT CUSPID, NVL(COMMPLAN, :DFLTCOMM) COMMPLAN
FROM ORD
```

The value of `DFLTCOMM` replaces null values of `COMMPLAN` in the rows selected.

**WHERE Clause**

```
SELECT ORCID, TOTAL
FROM ORD
WHERE CUSTID = :CUST
```

The value of `CUST` is used to select a single customer.

**GROUP BY Clause**

```
SELECT NVL (COMPLAN, :DFLTCOMM) COMMPLAN, SUM(TOTAL) TOTAL
FROM ORD
GROUP BY NVL(COMMPLAN, :DFLTCOMM)
```

All non-aggregate expressions such as `NVL (COMPLAN, :DFLTCOMM)` in the `SELECT` clause must be replicated in the `GROUP BY` clause.

**HAVING Clause**

```
SELECT CUSPID, SUM (TOTAL) TOTAL
FROM ORD
GROUP BY CUSTID
HAVING SUM (TOTAL) > :MINTOTAL
```

The value of `:MINTOTAL` is used to select customers with a minimum total of orders.

**ORDER BY Clause**

```
SELECT ORCID, SHIPDATE, ORDERDATE, TOTAL
FROM ORD
ORDER BY DECODE (:SORT, 1, SHIPDATE, 2, ORDERDATE)
```

The value of `:SORT` is used to select either `SHIPDATE` or `RDERDATE` as the sort criterion. Note that this is not the same as `ORDER BY 1` because `:SORT` is used as a value rather than to identify the position of an expression in the `SELECT` list.

**CONNECT BY and  
START WITH Clauses**

Parameters in `CONNECT BY` and `START WITH` clauses are used in the same way as they are in the `WHERE` and `HAVING` clauses.

## Creating Lexical Parameters

To create a lexical parameter, you must do the following:

1. Select Parameter from the Main Menu.
2. Use [Insert Record Above] or [Insert Record Below] to create a new record.
3. Enter a parameter name and also its corresponding attributes.
4. Assign an appropriate default value, if necessary or desired (see note below).
5. Use the parameter in a SELECT Statement, and precede the parameter with an ampersand (&).

**Note:** Lexical parameters must be specified in the Parameter Screen before you may reference them in a query, unless you have designed the query so that it will parse the query-even if the lexical parameter is null. In this case, "null" implies "blank," not the word "NULL."

For example, say you create the following two queries without first defining the lexical parameter &CUST (including its default value):

```
SELECT ORDID, TOTAL FROM ORD WHERE TOTAL >100 &CUST
```

SQL\*ReportWriter parses the query without error because the SELECT statement is correct if &CUST is replaced with a blank.

```
SELECT ORDID, TOTAL FROM ORD WHERE &CUST
```

SQL\*ReportWriter raises an error while parsing the query because the WHERE clause is not complete.

You can use lexical parameters to replace the clauses appearing after WHERE, GROUP BY, ORDER BY, HAVING, CONNECT BY, and START WITH. Examples of lexical parameters are found below:

### WHERE Clause

```
SELECT ORDID, TOTAL
FROM ORD
WHERE &CUST
```

&CUST can be used to restrict records retrieved from ORD. Any form of the WHERE clause can be specified at run-time.

### GROUP BY Clause

```
SELECT NVL (COMMPLAN, DFLTCOMM) CPLAN, SUM (TOTAL) TOTAL
FROM ORD
GROUP BY &NEWCOMM
```

The value of &NEWCOMM can be used to define the GROUP BY clause.

## HAVING Clause

```
SELECT CUSPID, SUM (TOTAL) TOTAL
FROM ORD
GROUP BY CUSTID
HAVING &MINTOTAL
```

The Value of &MINTOTAL could, for example, be used to select customers with a minimum total of orders.

## ORDER BY Clause

```
SELECT ORDID, SHIPDATE, ORDERDATE, TOTAL
FROM ORD
ORDER BY &SORT
```

The Value of &SORT Can be used to Select SHIPDATE, ORDERDATE, ORDID, Or any combination as the sort criterion. It could also be used to add on to the query, for example to add a "connect by" and "start with" clause.

## CONNECT BY and START WITH Clauses

Parameters in CONNECT BY and START WITH clauses are used in the same way as they are in the WHERE and HAVING clauses.

## Deleting a Parameter

To delete a bind or lexical parameter, you must delete the parameter from any text objects that reference it on the Text Screen, and delete the parameter from any SELECT statement on the Query Screen, and delete the parameter from the Parameter Screen. To delete a parameter from the Query or Text Screen, position your cursor on the first letter of the parameter and press [Delete Word]. To delete a parameter from the Parameter Screen, position your cursor on the name of the parameter and press [Delete Record].

Note: Deleting parameters does not delete references to them in texts. References to deleted parameters in text are treated as literal text. References to deleting parameters in queries will cause an error.

## Renaming a Parameter

Rename bind parameters by typing over the old name in the SELECT statement, in any text objects, and the Parameter Screen. To rename lexical parameters, you must first type over the old name in the Parameter Screen, and then change the name of the parameter in the SELECT statement and any text objects. If the old parameter is not deleted or renamed on the Query Screen, it will cause an error.

**Note:** If the old parameter name is not deleted or renamed on the Text Screen, it will be treated as literal text.

## Moving a Parameter

To move a bind or lexical parameter, edit the SELECT statement, delete the parameter reference from the text object (if the text object is edited), and add the parameter reference to the desired text object (if that text object is edited).

**Note:** Moving a bind or lexical parameter on the Parameter Screen will only move the position of the parameter on the Run-time Parameter Form. To move a parameter on the Parameter Screen, do the following

1. Place your cursor on the parameter field you want to move and press [Delete Record].
2. Move your cursor above the line where you want the deleted parameter field to appear and press [Undelete Record].

## Parameter Screen One

Parameter Screen One (Figure 10-1) is used to alter default values, and to create and define new parameters. The settings of this screen are

- Parameter Name
- Data Type
- Width
- Default Value
- Label.

**FIGURE 10-1**  
Parameter Screen One

| Parameter Name | Data Type | Width | Defaultt Value | Label |
|----------------|-----------|-------|----------------|-------|
|                |           |       |                |       |

Enter a name for this parameter. >

Report Name: <Replace>

**Parameter Name** The name of the parameter as used, or to be used, in either the SELECT statement or the text object.

**Default** When a parameter is created by default in a query, the name used in the query.

**Options** Modify specifications of an existing parameter.

- Rules**
1. Parameter names must be distinct from all other report objects, and must conform to SQL naming standards.
  2. Parameter names must not be the same as any SQL\*ReportWriter object name or RUNREP command line keywords. Reserved command line keywords are:

|           |           |          |
|-----------|-----------|----------|
| ARRAYSIZE | DESFORMAT | READONLY |
| BATCH     | DESNAME   | REPORT   |
| BUFFERS   | DESTYPE   | TERM     |
| CMDFILE   | LOGFILE   | USERID   |
| COPIES    | PARAMFORM |          |

Bind parameters must not be the same as any reserved SQL keywords. For a list of reserved SQL keywords, consult the *SQL Language Reference Manual*.

## Data Type

Specifies the datatype for the parameter field.

**Default** CHAR

**Options** *CHAR* indicates the parameter value may be any valid characters.

*NUM* indicates the parameter value must be a number.

*DATE* indicates the parameter value must be a date.

- Rules**
1. Values supplied at run-time must conform to the datatype specified on this screen.
  2. For reports that run against ORACLE databases, you may ignore this setting because ORACLE converts the datatype automatically. However, after the conversion ORACLE does not automatically use indexes for those queries.

For other databases, the correct datatype is usually required for comparisons and functions to work properly. However, if you want to validate using input masks, you must specify the datatype.

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Width</b>             | The maximum number of characters for the parameter value that will be used as input for the Run-time Parameter Form and report output.                                                                                                                                                                                                                                                                                                                                                              |
| <b>Default</b>           | 40                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Options</b>           | Enter any value equal to or between 1 and 999.'                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Rules</b>             | SQL*ReportWriter truncates any parameter value passed at run-time that is longer than the width you specify.                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Default Value</b>     | Specifies` the initial value of the parameter. This value is used unless it is overridden on the command line or in the run-time parameter form.                                                                                                                                                                                                                                                                                                                                                    |
| <b>Default</b>           | Blank, implying no default value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Options</b>           | Enter a value appropriate for the datatype. The current date maybe used as a default value. To do so, you must assign the default value of &DATE to a parameter with a datatype of <i>DATE</i> .                                                                                                                                                                                                                                                                                                    |
| <b>Rules</b>             | <ol style="list-style-type: none"> <li>1. The default value is validated according to the datatype and width.</li> <li>2. The value can have a maximum length of 240 characters.</li> </ol>                                                                                                                                                                                                                                                                                                         |
| <b>Label</b>             | Specifies the text that appears as a prompt on the run-time parameter form.                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Default</b>           | The default parameter label is the text of the parameter name, modified following the same rules used for generating field labels (see Chapter 8, "Field Screens").                                                                                                                                                                                                                                                                                                                                 |
| <b>Options</b>           | You can change the label by typing over the default value.                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Rules</b>             | The maximum width of a Label is 27 characters.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>System Parameters</b> | <p>The following are the system parameters that are found on the parameter screen. They cannot be deleted, but you can re-order them, relabel them, suppress their display on the Run-time Parameter Form (using SKIP), and change their default values to modify the display of the Run-time Parameter Form.</p> <ul style="list-style-type: none"> <li>• DESTYPE</li> <li>• DESNAME</li> <li>• DESFORMAT</li> <li>• COPIES</li> <li>• CURRENCY</li> <li>• DECIMAL</li> <li>• THOUSANDS</li> </ul> |

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DESTYPE</b>   | Specifies the type of device to which to send report output.                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Default          | Screen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Options          | Screen sends report output to the screen.<br>File sends report output to the specified file.<br><i>Printer</i> sends report output to the printer named in the next settings.<br>Mail sends report output to Oracle*Mail users. When this occurs, a report will be sent as an attached file, not directly as a mail message. You may, therefore, use any printer description file when executing the report.<br>Sysout sends report output to "Standard Output" for the RUNREP command. |
| Rules            | You can only use Sysout as a DESTYPE option when you are running reports via RUNREP with the BATCH = YES keyword.                                                                                                                                                                                                                                                                                                                                                                       |
| <b>DESNAME</b>   | The printer name, if DESTYPE (above) is <i>Printer</i> ; filename if DESTYPE is <i>File</i> ; Oracle*Mail user if DESTYPE is Mail; otherwise ignored.                                                                                                                                                                                                                                                                                                                                   |
| Default          | <report name>.lis (this name may vary by operating system).                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Options          | See your System Administrator for the names of supported printers.                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>DESFORMAT</b> | Specifies the characteristics of the printer, file, or Oracle*Mail message named in DESNAME, to which the report output will be routed. This keyword is ignored when the DESTYPE is Screen.<br><br>See your System Administrator for a list of valid formats.                                                                                                                                                                                                                           |
| <b>COPIES</b>    | Specifies the number of copies of the report to be printed.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Default          | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Options          | Enter any positive, whole number.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Rules            | If the DESTYPE option is specified to be Printer (above), then this option may behave differently on different operating systems. Whenever multiple copies are specified, SQL*ReportWriter relies on the operating system's PRINT spooling facility to print the required number of copies. This keyword is ignored for any DESTYPE other than <i>Printer</i> .                                                                                                                         |

**CURRENCY**

The symbol that is used for the currency indicator.

**Default** Database dependent for ORACLE Version 6.0; "\$" for previous versions of ORACLE.

**Options** Enter any symbol (without quotes).

**Rules** An entry for this system parameter will override any made with the `LANGUAGE=VALUE` command-line argument.

**THOUSANDS**

The string that is used for the thousands indicator.

**Default** Database dependent for ORACLE Version 6.0;"," for previous versions of ORACLE.

**Options** Enter any string (without quotes).

**Rules** An entry for this system parameter will override any made with the `LANGUAGE=VALUE` command-line argument.

**DECIMAL**

The symbol that is used for the decimal indicator.

**Default** Database dependent for ORACLE Version 6.0; "." (a period) for previous versions of ORACLE.

**Options** Enter any symbol (without quotes).

**Rules** An entry for this system parameter will override any made with the `LANGUAGE=VALUE` command-line argument.

  
**Parameter Screen Two**

Parameter Screen Two (Figure 10-2) is used to specify the input and output appearance of parameter values that you input. The settings of this screen include

- Input Format
- Output Format
- Skip

**FIGURE 10-2**  
**Parameter Screen Two**

| Action                                                                                                                                       | Query | Group | Field          | Summary      | Text          | Report | Parameter | Help |
|----------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|----------------|--------------|---------------|--------|-----------|------|
| Parameter Settings                                                                                                                           |       |       |                |              |               |        |           |      |
|                                                                                                                                              |       |       | Parameter Name | Input Format | Output Format | Skip   |           |      |
| Enter the Format mask (e.g. \$999.99) to be used on the parameter form.<br>Report Name: <span style="float: right;">&lt; Replace &gt;</span> |       |       |                |              |               |        |           |      |

**Input Format**

Specifies the Display Format to use for the parameter when it appears on the Run-time Parameter Form. It is also used to validate the default value of the parameter.

- Default Blank. (When the datatype is *DATE*, a null Input Format implies 'DD-MON-YY'.)
- Options Any valid Display Format. See "Display Formats" of Field Screen Two, or press [Help].

**Output Format**

Specifies the display format to use when displaying the parameter in the report output.

- Default Blank. (When the datatype is *DATE*, a null Output Format implies 'DD-MON-YY'.)
- Options Any valid Display Format. See "Display Formats" in the Field Screen Two section.

**Skip**

Specifies that SQL\*ReportWriter should not display the indicated parameter on the Run-time Parameter Form. This is useful in a turnkey or production system where the operator is to have access to only a subset of the report's parameters.

- Default Blank, implying that the parameter may be edited on the Run-time Parameter Form.
- Options X suppresses the display of the parameter on the Run-time Parameter Form.

Rules If all parameters are skipped in a report, the Run-time Parameter Form will not appear when the report is executed.

# A

## KEY COMMANDS

This appendix contains information about the SQL\*ReportWriter function keys. The keys are divided into three categories

- text editing keys
- navigation keys
- global keys.

For the actual keyboard mappings, see your system-specific *Installation and Users' Guide*.

## Text Editing Keys

|                   |                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------|
| Right             | Moves the cursor one character to the right.                                                |
| Left              | Moves the cursor one character to the left.                                                 |
| Delete Character  | Deletes the character at the cursor.                                                        |
| Delete Backward   | Deletes the first character to the left of the cursor.                                      |
| End of Line       | Moves the cursor to the end of the current line.                                            |
| Beginning of Line | Moves the cursor to the beginning of the current line.                                      |
| Delete Line       | Deletes the entire line.                                                                    |
| Delete to BOL     | Deletes the portion of the line from the cursor to the beginning of the line.               |
| Delete to EOL     | Deletes the portion of the line after the cursor.                                           |
| Undelete' Line    | Undeletes the last line that has been deleted.                                              |
| Insert Line       | Inserts anew line beneath the current line.                                                 |
| Previous Line     | Positions the cursor on the previous line at the same column position it was previously in. |
| Next Line         | Positions the cursor on the next line at the same column position it was previously in.     |
| First Line        | Positions the cursor on the first line.                                                     |
| Last Line         | Positions the cursor on the last line.                                                      |
| Add Newline       | Adds a blank line before the current line.                                                  |
| Add Tab           | Moves the cursor to the next tab.                                                           |
| Delete Word       | Deletes the word to the right of the cursor.                                                |
| Undelete Word     | Undeletes the last word that has been deleted.                                              |
| Next Word         | Moves the cursor to the first character of the first word to the right of the current word. |
| Previous Word     | Moves the cursor to the first character of the first word to the left of the current word.  |

|                    |                                                                                                                                                                                                                                                  |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Zoom               | Expands the space occupied by the current field for easier editing. To return the field to its normal size, press any key that would normally take you out of that field ([Next Field], [Previous Field], [Accept], etc).                        |
| Read File          | Reads a file into the current field.                                                                                                                                                                                                             |
| Write File         | Writes the contents of the buffer into an external text file. The file is created if it doesn't exist, and overwritten if it does exist.                                                                                                         |
| Search             | Places the cursor on the next instance of the specified text in the current field.                                                                                                                                                               |
| Search and Replace | Finds the next instance of the specified text in the current field and replaces it with the new specified text.                                                                                                                                  |
| Insert/Replace     | Toggles between insert and replace text modes.                                                                                                                                                                                                   |
| Mark               | Marks the beginning point of text for an editing operation such as copy, delete, or highlight. When the cursor is moved from the beginning point, the text will appear in reverse-video from the beginning point to the current cursor position. |
| Cut                | Removes a marked section of text and places it in the buffer; this key is used in conjunction with the Paste and Write File keys.                                                                                                                |
| Copy               | Copies a section of marked text into the buffer without affecting the field you are editing; this key is used in conjunction with the Paste and Write File keys.                                                                                 |
| Paste              | Places text from the buffer into the report, starting at the cursor.                                                                                                                                                                             |
| Highlight          | Invokes a menu from which you can choose a highlighting style; this key is used in conjunction with the Mark key. Highlight is only available in the text area of the Text Screen.                                                               |
| Ignore             | Ignores a specified sequence of keystrokes.                                                                                                                                                                                                      |

## Navigation Keys

|                                       |                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Next Field                            | Moves the cursor forward to the next field. In the last field of a dialog box or the Run-time Parameter Form, [Next Field] closes the window and completes the operation.                                                                                                                                                                                                |
| Previous Field                        | Moves the cursor back to the previous field.                                                                                                                                                                                                                                                                                                                             |
| First Field                           | Moves the cursor back to the first field of the current record.                                                                                                                                                                                                                                                                                                          |
| Last Field                            | Moves the cursor to the last field of the current record.                                                                                                                                                                                                                                                                                                                |
| Insert Rec. Below                     | Inserts a record below the current record.                                                                                                                                                                                                                                                                                                                               |
| Insert Rec. Above                     | Inserts a record above the current record.                                                                                                                                                                                                                                                                                                                               |
| Delete Record                         | Deletes the current record.                                                                                                                                                                                                                                                                                                                                              |
| Undelete Record                       | Undeletes the last record you deleted. Undelete is only available on the Group, Field, Summary, and Parameter Screens.                                                                                                                                                                                                                                                   |
| Previous Record                       | Moves the cursor to the previous record.                                                                                                                                                                                                                                                                                                                                 |
| Next Record                           | Moves the cursor to the next record.                                                                                                                                                                                                                                                                                                                                     |
| First Record                          | Moves the cursor to the first record.                                                                                                                                                                                                                                                                                                                                    |
| Last Record                           | Moves the cursor to the last record.                                                                                                                                                                                                                                                                                                                                     |
| Scroll Left/Right or scroll Prev/Next | On field, group, and summary screens, displays the screen to the left/right of the current screen. On the query screen and the text screen, displays the previous/next query or text object.                                                                                                                                                                             |
| Scroll Up/Down                        | On field, group, and summary screens, and List of values, displays the set of records above/below the current screen. On the query screen and the text screen, displays the prior/next query or text object; in the SELECT statement, Comment, and Text settings, scrolls up/down within that setting. In the browser, scrolls up/down by page of output, not by screen. |

|                               |                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scroll Leftmost/<br>Rightmost | On field, group, and summary screens, displays the first/last screen. On the query screen and the text screen, displays the first/last query or text object.                                                                                                                                                                                                                                  |
| Scroll Top/Bottom             | On field, group, and summary screens, and List of values, displays the first/last record. On the query screen and the text screen, displays the first/last query or text object; in the SELECT statement, Comment, and Text settings, Scrolls to the first/last line of text. In the browser, scrolls to the first line of the first page (Top), or the first line of the last page (Bottom). |
| Help Scroll Up/Down           | Scrolls by page within a help topic.                                                                                                                                                                                                                                                                                                                                                          |
| Help Scroll Top/Bottom        | Displays the first/last page of a help topic.                                                                                                                                                                                                                                                                                                                                                 |
| Window                        | Moves the cursor to the bottom of the screen to allow for line-by-line and character-by-character scrolling of report output. Use Window mode to view reports whose page dimensions are larger than the screen. Window is only available when browsing report output on the screen.                                                                                                           |
| First Choice                  | Moves the cursor to the first item in a List of values. (Mapped from First Record.)                                                                                                                                                                                                                                                                                                           |
| Last Choice                   | Moves the cursor to the last item in a List of values. (Mapped from Last Record.)                                                                                                                                                                                                                                                                                                             |
| Next Choice                   | Moves the cursor to the next item in a menu or List of values.                                                                                                                                                                                                                                                                                                                                |
| Previous Choice               | Moves the cursor to the previous item in a menu or List of values.                                                                                                                                                                                                                                                                                                                            |
| Menu                          | Places the cursor in the Main Menu without refreshing the screen. If you leave a definition screen with [Menu] and then execute a report, upon completion you will return to the screen you left.                                                                                                                                                                                             |
| List                          | Causes a window to appear with a list of values for the current context when the possible values aren't already present on your screen.                                                                                                                                                                                                                                                       |
| Select                        | Selects a menu choice or a value from a List of values.                                                                                                                                                                                                                                                                                                                                       |

## Global Keys

|               |                                                                                                                                                                                                                                                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Accept        | Commits changes to report definition screens and places the cursor in the Main Menu. Also validates information on the run-time parameter form, thus allowing the execution of a report to begin; selects choice from List of values, and exits dialog boxes.                                                |
| Undo          | Exits definition screens without saving changes; exits menus, dialog boxes, and alert boxes without selecting an option. In the run-time parameter form, stops the report from executing. In the help system, backtracks one screen at a time to all the screens the user has viewed in the current session. |
| Query         | Invokes query mode on the Text Screen so you can specify objects to be retrieved, based on their settings; used in conjunction with Fetch.                                                                                                                                                                   |
| Fetch         | Executes a query and fetches objects with matching settings; used in conjunction with Query. If no settings are specified, retrieves all objects.                                                                                                                                                            |
| Help          | Brings up a context-sensitive help screen.                                                                                                                                                                                                                                                                   |
| Key Help      | Displays a list of key commands for the current terminal definition file.                                                                                                                                                                                                                                    |
| Refresh       | Refreshes the screen.                                                                                                                                                                                                                                                                                        |
| Bookmark      | Marks a place in the help system which allows you to toggle back and forth between the SQL*ReportWriter screens and that marked screen. Each screen from which you invoke Bookmark subsequently becomes the currently marked screen                                                                          |
| Repeat        | Used in conjunction with another function key; allows you to repeat the function you specify a certain number of times.                                                                                                                                                                                      |
| Shell to Host | Takes you into a subprocess that acts like your system prompt. When you exit this subprocess, you return to the place in SQL*ReportWriter that you left.                                                                                                                                                     |

|                    |                                                                                                                                                                                                                                                                                                                                                             |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cancel Application | Exits you immediately from SQL*ReportWriter.                                                                                                                                                                                                                                                                                                                |
| Print Page         | Appends the current page to the logfile.                                                                                                                                                                                                                                                                                                                    |
| Print Screen       | Appends the current screen to the logfile.                                                                                                                                                                                                                                                                                                                  |
| Print Report       | In the output screen, appends the entire report output to the logfile. In the query, group, field, summary, text, report, and parameter screens, appends to the logfile all of the object screens associated with the current object screen (for example, if Field Screen One is the current screen, all of the field screens are appended to the logfile). |

# B

## ADMINISTRATION

### Text

**T**his appendix contains instructions for:

- upgrading from previous versions of SQL\*ReportWriter
- installing a userid
- dropping reports
- preparing for winding report outputs via Oracle\*Mail
- establishing security and restrictions.

This appendix also contains information on:

- memory and disk space requirements of SQL\*ReportWriter
- database storage requirements
- SQL\*ReportWriter's file-searching method
- files that maybe opened during run-time.

Finally, this appendix discusses how to document your reports using reports provided with SQL\*ReportWriter.

## Upgrading from Previous Versions

To upgrade from a previous version of SQL\*ReportWriter to Version 1.1, you need to run the MOVEREP program after satisfying the following:

- When using ORACLE RDBMS Version 6.0 with the transaction processing option, you need SQL\*ReportWriter Version 1.0 tables in your account, and SQL\*ReportWriter Version 1.1 tables in the SYSTEM account (and the Database Administrator must have granted you access to it)
- When using any other version of ORACLE, you need the SQL\*ReportWriter Version 1.0 tables and SQL\*ReportWriter Version 1.1 tables in your account.

To run MOVEREP, enter the following at the system prompt:

```
moverep [USERID=] userid
```

where:

*USERID*                    the ORACLE username/password@node of your  
(not SYSTEM) account.

To upgrade from previous versions of SQL\*ReportWriter and/or ORACLE, refer to the chart on the following page.

Note: When moving to a new release of SQL\*ReportWriter (for example, moving from Version 1.1.8 to Version 1.1.10), you may need to regenerate your rep files.

|                      | SRW1.0<br>ORACLE V5 | 1.0<br>V6        | 1.1<br>V5           | 1.1<br>V6               | 1.1<br>V6 *                      |
|----------------------|---------------------|------------------|---------------------|-------------------------|----------------------------------|
| SRW 1.0<br>ORACLE V5 | ---                 | exp/<br>imp      | move rep            | exp/imp<br>then moverep | exp/imp<br>then moverep          |
| 1.0<br>V6            | dumprep/<br>loadrep | ---              | **                  | moverep                 | dumprep/ loadrep<br>then moverep |
| 1.1<br>V5            | not<br>supported    | not<br>supported | ---                 | dumprep/<br>loadrep     | dumprep/<br>loadrep              |
| 1.1<br>V6            | not<br>supported    | not<br>supported | dumprep/<br>loadrep | ---                     | dumprep/<br>loadrep              |
| 1.1<br>V6"           | not<br>supported    | not<br>supported | dumprep/<br>loadrep | dumprep/<br>loadrep     | ---                              |

SRW SQL\*ReportWriter

\* ORACLE RDBMS Version 6.0 with the transaction processing option

\*\* export the SQL\*ReportWriter 1.0 tables through SQL\*Net, then import them into ORACLE V5, then use moverep

**exp/imp** export the SQL\*ReportWriter tables, then import them

**dumprep/loadrep** use the dumprep command, then loadrep command

## Installing a Userid

To install a new ORACLE userid after the initial installation of SQL\*ReportWriter, do the following using SQL\*Plus

1. A user with DBA privileges must issue the following commands:

```
GRANT RESOURCE, CONNECT TO <userid>
IDENTIFIED BY <password>
CONNECT <userid/password>
```

Where <userid> is the new userid, and <password> is the new password.

2. On ORACLE with the transaction processing option, run the *srw\_grnt.sql* script to grant privileges to the user. Revoke privileges by running *srw\_rvke.sql*.
3. On other ORACLE databases, you must install tables for that user with the *srw\_iloc.sql2* script.

---

## Dropping Reports

Reports can be dropped by running the `srw_drop.sql` script. If you want to drop more than one report, you can use the percent sign (%) and underscore (\_) as wildcards, as you would in a SQL LIKE expression. Only reports created by you will be dropped, even if you have been granted access to others' reports.

---

## Preparing for Sending Reports via Oracle\*Mail

When a user specifies *Mail* as the destination type (DESTTYPE) from SQLREP or RUNREP, SQL\*ReportWriter invokes a command file, `srw_mail.com`, to call the Oracle\*Mail ORASEND utility. The Database Administrator should modify this command file to 1) set up the Oracle\*Mail database before the ORASEND call is made, and 2) reset to the SQL\*ReportWriter database on return from ORASEND. Note: Although the command file may be altered to call any mail system, mailing through any system other than Oracle\*Mail is not supported.

---

## Establishing Security and Restrictions

There are three types of restrictions that can be placed on users of SQL\*ReportWriter:

- . preventing users from running SQL\*ReportWriter
- | preventing users from accessing other users' reports
- | limiting the maximum number of output pages for each user.

SQL\*ReportWriter's security mechanism supplements ORACLE's GRANT and REVOKE commands, providing extra measures of control. All restrictions placed through GRANT and REVOKE still apply.

## Controlling Access to SQL\*ReportWriter

If you installed system-owned SQL\*ReportWriter tables, where users share one set of SQL\*ReportWriter tables, all users will be able to access SQL\*ReportWriter. (The `srw_icen.sql` procedure installs system-owned tables.) Use `srw_grnt.sql` or `srw_rvke+I` to grant or revoke privileges of users for SQL\*ReportWriter.

If you installed user-owned SQL\*ReportWriter tables, where all users have their own set of SQL\*ReportWriter tables, a user will be unable to run SQL\*ReportWriter unless tables are created for that user by running the `srw_iloc.sql` script.

## Controlling Access to Other Users' Reports

If you installed system-owned SQL\*ReportWriter tables, where users share one set of SQL\*ReportWriter tables, users are only able to access another user's report if the report owner has entered his or her name (or PUBLIC) in the Access field on the Report Screen. If the owner enters PUBLIC, all users will have access to the report. Users cannot change the Access list of other users' reports.

If you installed user-owned SQL\*ReportWriter tables, report-level security is not supported. To share a single report, use DUMPREP or LOADREP. To allow/prevent another user to copy all of your reports, run the `snu_gmf.sql` or `srw_rukesq2` script. The grantee will then be able to copy reports using SQLREP's Copy choice on the Action menu.

## Limiting the Number of Pages of Output

The DBA can limit the number of pages of output that a given user can generate by running the `srw_2mt.sql` script (see your *ORACLE Installation Guide* to learn where the file is stored). With this script, the DBA sets a maximum page number; the default limit is 0, which implies that the number of pages is unlimited. Doing this will insert a row in the PRODUCT\_PROFILE table. See Appendix C for the table structure.

SQL\*ReportWriter reads the restrictions from the PRODUCT\_PROFILE table when a user logs onto SQL\*ReportWriter and maintains those restrictions for the duration of the session. Therefore, changes to the PRODUCT\_PROFILE table will only take effect when the affected user logs onto SQL\*ReportWriter.

## Memory and Disk Space Requirements

SQL\*ReportWriter has the following memory requirements:

| <i>Item</i>               | <i>Memory</i> | <i>Disk Space</i> |
|---------------------------|---------------|-------------------|
| Run-time engine           | 0.5meg        | 4.0 meg           |
| Per user process          | 0.5meg        |                   |
| Define-time engine shared | 1.0 meg       | 6.0 meg           |
| Per user process          | 0.5meg        |                   |

**Note:** Because this information can vary from system to system and across dissimilar hardware, the table above is only provided to give you a general idea of memory requirements.

## **Extra Disk Space**

Be aware that temporary work files created while running a report can take up a large amount of disk space, much more than the amount required for the final report. Since the amount of disk space used varies from report to report, make sure you have enough extra hard disk space available when running your reports.

## **Database Storage Requirements**

SQL\*ReportWriter tables require approximately 100 database blocks of storage when empty. When using SQL\*ReportWriter with an ORACLE transaction processing option database, only one set of tables is created and therefore only 100 blocks are needed initially per installation. On all other ORACLE databases, each user has his or her own set of tables. Consequently, approximately 100 database blocks are needed per user. As reports are built, more storage maybe needed.

The storage occupied by the tables increases as reports are defined. The actual storage per report can vary significantly, depending on the amount of text entered in the Text and Query Screens, and also on the number of queries. However, it has been our experience that reports typically occupy 3-6 database blocks each.

## **File-Searching Method**

SQL\*ReportWriter first looks to see if a filename and path has been provided by the user, if so, it looks there. Then, if no filename has been provided, SQL\*ReportWriter looks for the default filename and path (system-dependent, see your Installation and User's Guide). Lastly, if SQL\*ReportWriter cannot find the file, an error is raised.

## Files Opened During Run-time

The following files may be opened at some time during run-time

| Filename and Function                        | <i>SQLREP</i>                                 | <i>RUNREP</i>               | <i>GENREP</i>       | <i>DUUPREP</i> | <i>LOADREP</i> | <i>PRINTDEF</i> |
|----------------------------------------------|-----------------------------------------------|-----------------------------|---------------------|----------------|----------------|-----------------|
| SQLREP file<br>sr.a                          | *ALWAYS                                       | --                          | --                  | --             | --             | --              |
| help file<br>sr.h                            | ALWAYS                                        | WHEN<br>BATCH=NO            | --                  | --             | --             | --              |
| report description file<br><report name>.rep | GENERATE &<br>EXECUTE                         | ALWAYS                      | ALWAYS              | --             | --             | --              |
| temporary log file<br><temp>                 | ALWAYS                                        | ALWAYS                      | --                  | --             | --             | --              |
| report output file<br><report name>.lis      | GENERATE &<br>EXECUTE,<br>DESTYPE=<br>FILE    | WHEN<br>DESTYPE=<br>FILE    | --                  | --             | --             | --              |
| printer output file                          | GENERATE &<br>EXECUTE,<br>DESTYPE=<br>PRINTER | WHEN<br>DESTYPE=<br>PRINTER | --                  | --             | --             | --              |
| terminal description file                    | ALWAYS                                        | WHEN<br>BATCH=NO            | --                  | --             | --             | --              |
| printer description file<br><Cub             | GENERATE &<br>EXECUTE                         | ALWAYS                      | --                  | --             | --             | ALWAYS          |
| load file<br><filename>.rex                  | --                                            |                             | WHEN USING<br>FILE= | ALWAYS         | ALWAYS         | --              |
| printer definition file<br><printdef.dat>    |                                               |                             |                     |                |                | ALWAYS          |

\* System dependent: on most large systems the sr. a file is not present.

## **Documenting Your Reports**

You can print the SQL\*ReportWriter report settings for any of your reports using one of the following reports shipped with SQL\*ReportWriter

- SRW80
- SRW132
- SRW175.

The number in the report names indicates the width of the report output in characters. The report settings displayed are the same for all three pre-packaged reports, from the most global settings to the most detailed.

When you run one of the reports, you can choose to document a single report by typing its name on the first line in the Run-time Parameter Form, or you can use the wildcard value (%) to specify a subset of your reports. The default is% alone, which produces a report on all of your reports.

## **Accessing System-owned Tables**

When using the system-owned SQL\*ReportWriter tables, only one user may update the system tables at any given time. Thus, if one user invokes a SQL\*ReportWriter executable (SQLREP, MOVEREP, etc.) that updates the system-owned tables, SQL\*ReportWriter will prevent other users from invoking an executable that may update the system tables until the first user's updates have been completed. For example, if a user is using DUMPREP, another user will be unable to open the same report using RUNREP or SQLREP.

# C

## SQL\*REPORTWRITER TABLES

This appendix contains a list of the SQL\*ReportWriter tables. There are two ways in which to install the SQL\*ReportWriter tables: 1) to install system-owned SQL\*ReportWriter tables, where users share one set of SQL\*ReportWriter tables; and 2) to install user-owned SQL\*ReportWriter tables, where all users own their own set of SQL\*ReportWriter tables. System-owned tables are possible only with the ORACLE RDBMS Version 6.0 with the transaction processing option.

Because the SQL\*ReportWriter tables are different based on how they are installed, this appendix contains three sections

- tables that are used when either system-owned or user-owned SQL\*ReportWriter tables are installed
- additional database objects that are available when system-owned SQL\*ReportWriter tables are installed
- additional database objects that are used when user-owned SQL\*ReportWriter tables are installed.

This appendix is provided for informational purposes only. Do not alter the contents of these tables through SQL\*Plus or other means because you may destroy or invalidate your report definitions by doing so.

## Tables Used for both System-owned and User-owned SQL\*ReportWriter Tables

| SRW_FIELD | Name            | Null?    | Type      | comment                                                                                                                         |
|-----------|-----------------|----------|-----------|---------------------------------------------------------------------------------------------------------------------------------|
|           | ALIGNMENT       | NULL     | NUMBER(1) | Field Alignment<br>1 =Left, 2=Right, 3=center,<br>4=Wrap, 5=Variable                                                            |
|           | APPID           | NOT NULL | NUMBER(9) | Report Id                                                                                                                       |
|           | COMPUTE         | NULL     | CHAR(240) | User Exit Text or<br>System Variable Name                                                                                       |
|           | DATATYPE        | NULL     | NUMBER(1) | Datatype<br>1=NUM, 2=CHAR/<br>3=DATE, 4=PRT                                                                                     |
|           | FIELD_ORDER     | NULL     | NUMBER(3) | Order of Field on<br>Field Screen                                                                                               |
|           | FORMAT_MASK     | NULL     | CHAR(40)  | Display Format                                                                                                                  |
|           | GROUPID         | NOT NULL | NUMBER(9) | Group Id                                                                                                                        |
|           | HEADING         | NULL     | CHAR(240) | Field Label                                                                                                                     |
|           | ITEMID          | NOT NULL | NUMBER(9) | Field Id                                                                                                                        |
|           | LINES_BEFORE    | NULL     | NUMBER(3) | Lines Before                                                                                                                    |
|           | OPERATOR        | NULL     | NUMBER(2) | Computed Field Functions<br>1=sum, 2=Min, 3=Max,<br>4=Count, 5=Avg<br>6=%Total, +50 for<br>Running Functions<br>7=First, 8=Last |
|           | OWNER           | NULL     | CHAR(30)  | Report Owner                                                                                                                    |
|           | RELATIVE_POS    | NULL     | NUMBER(1) | Relative Position of Field<br>1=Below, 2=Right,<br>3=Panel                                                                      |
|           | REPRINT         | NULL     | CHAR(1)   | Display on All Panels<br>of Group<br>X=Yes, Null=No                                                                             |
|           | RESET_GROUP     | NULL     | NUMBER(9) | Reset Group for<br>Computed Field                                                                                               |
|           | SKIP            | NULL     | CHAR(1)   | skip:<br>X=Skip, Null=Print                                                                                                     |
|           | SOURCE_QUERY    | NULL     | NUMBER(9) | Query Of Source Column                                                                                                          |
|           | SPACES_BEFORE   | NULL     | NUMBER(3) | Spaces Before                                                                                                                   |
|           | TARGET_POSITION | NULL     | NUMBER(3) | select List Position<br>in Query                                                                                                |
|           | WIDTH           | NULL     | NUMBER(3) | Physical Field Width                                                                                                            |

**SRW\_FKEY**

| Name          | Null?    | Type      | Comment                                              |
|---------------|----------|-----------|------------------------------------------------------|
| APPID         | NOT NULL | NUMBER(9) | Report Id                                            |
| FOREIGN_TLPOS | NULL     | NUMBER(3) | Position of Column in<br>Select List of Parent Query |
| ITEMID        | NOT NULL | NUMBER(9) | Child Query Id                                       |
| LOCAL_TLPCB   | NULL     | NUMBER(3) | Position of Column in<br>Select List of Child Query  |
| OWNER         | NULL     | CHAR(W)   | Report Owner                                         |
| PARENTID      | NULL     | NUMBER(9) | Parent Query Id                                      |

**SRW\_GROUP**

| Name          | Null?    | Type      | Comment                                                                                                                           |
|---------------|----------|-----------|-----------------------------------------------------------------------------------------------------------------------------------|
| APPID         | NOT NULL | NUMBER(9) | Report Id                                                                                                                         |
| FIELDS_ACROSS | NULL     | NUMBER(3) | Maximum Fields Across                                                                                                             |
| FIELD_HILITE  | NULL     | NUMBER(2) | Field Highlight Style<br>0=Normal, 2=Underline,<br>4=Rev, 16=Bold 6=Und/Rev,<br>18=Und/Bold 20=Rev/Bold,<br>22=Und/Rev/Bold       |
| GROUP_ORDER   | NULL     | NUMBER(3) | Order on Screen                                                                                                                   |
| INTER_FIELD   | NULL     | NUMBER(3) | Spaces Between fields                                                                                                             |
| INTER_ROW     | NULL     | NUMBER(3) | Spaces Between Rows                                                                                                               |
| ITEMID        | NOT NULL | NUMBER(9) | Group Id                                                                                                                          |
| LABEL_HILITE  | NULL     | NUMBER(2) | Label Highlight Style<br>0=Normal, 2=Underline,<br>4=Rev, 16=Bold,<br>6=Und/Rev,<br>18=Und/Bold, 20=Rev/Bold,<br>22= Und/Rev/Bold |
| LINES_BEFORE  | NULL     | NUMBER(3) | Lines Before Group                                                                                                                |
| LOCATE_LABELS | NULL     | NUMBER(1) | Position of Field Labels<br>for this Group:<br>1 =Above, 2=Left                                                                   |
| MATRIX_FLAG   | NULL     | CHAR(1)   | Matrix Group Indicator:<br>X=Yes, Null=No                                                                                         |
| MULTI_PANEL   | NULL     | CHAR(1)   | Span Multiple Panels<br>X=Yes, Null=No                                                                                            |
| OWNER         | NULL     | CHAR(30)  | Report Owner                                                                                                                      |
| PAGE_BREAK    | NULL     | NUMBER(1) | Page Break Style<br>1=Conditional, 2=Always                                                                                       |
| QUERYID       | NULL     | NUMBER(9) | Query for this Group                                                                                                              |
| RELATIVE_POS  | NULL     | NUMBER(1) | Relative Position of Group<br>1=Below, 2=Right                                                                                    |
| REPETITION    | NULL     | NUMBER(1) | Repetition Direction                                                                                                              |
| SPACES_BEFORE | NULL     | NUMBER(3) | Spaces Before Group                                                                                                               |

**SRW\_PARAM**

| <i>Name</i>   | <i>Null?</i> | <i>Type</i> | <i>Comment</i>                               |
|---------------|--------------|-------------|----------------------------------------------|
| APPID         | NOT NULL     | NUMBER(9)   | Report Id                                    |
| DATATYPE      | NULL         | NUMBER(1)   | Parameter Datatype<br>1=NUM,<br>2=CHAR3=DATE |
| DEFAULT.VALUE | NULL         | CHAR(M)     | Initial Value                                |
| INPUT_MASK    | NULL         | CHAR(40)    | input Format Mask                            |
| ITEMID        | NOT NULL     | NUMBER(9)   | Parameter Id                                 |
| LABEL         | NULL         | CHAR(240)   | Parameter Label                              |
| OUTPUT_MASK   | NULL         | CHAR(a)     | Output Format Mask                           |
| OWNER         | NULL         | CHAR(30)    | Report Owner                                 |
| PARAM_ORDER   | NULL         | NUMBER(3)   | Order on Parameter Screen                    |
| PARAM_TYPE    | NULL         | NUMBER(2)   | Type:<br>0=System 1=Query, 2 = Text          |
| SKIP          | NULL         | CHAR(1)     | Skip on Parameter Form<br>X=Yes, Null=No     |
| WIDTH         | NULL         | NUMBER(3)   | Parameter Width                              |

**SRW\_QUERY**

| <i>Name</i>     | <i>Null?</i> | <i>Type</i> | <i>Comment</i>        |
|-----------------|--------------|-------------|-----------------------|
| APPID           | NOT NULL     | NUMBER(9)   | Report Id             |
| ITEMID          | NOT NULL     | NUMBER(9)   | Query Id              |
| MATRIX_PARENTID | NULL         | NUMBER(9)   | Parent Query 2 ID     |
| OWNER           | NULL         | CHAR(W)     | Report owner          |
| PARENTID        | NULL         | NUMBER(9)   | Parent Query ID       |
| QUERY           | NULL         | LONG        | SELECT Statement Text |
| QUERY_ORDER     | NULL         | NUMBER(3)   | order on screen       |

**SRW\_REPORT**

| <i>Name</i>      | <i>Null?</i> | <i>Type</i> | <i>Comment</i>                                 |
|------------------|--------------|-------------|------------------------------------------------|
| APPID            | NOT NULL     | NUMBER(9)   | Report Id                                      |
| BOTTOM_MARGIN    | NULL         | NUMBER(3)   | Bottom Margin                                  |
| COMMENTS         | NULL         | LONG        | comments                                       |
| CREATE_DATE      | NULL         | DATE        | Date Created                                   |
| LEFT_MARGIN      | NULL         | NUMBER(3)   | Left Margin                                    |
| MODIFIED_DATE    | NULL         | DATE        | Date Last Modified                             |
| MODIFIED_VERSION | NULL         | NUMBER(S)   | SQL*ReportWriter Version<br>When Last Modified |
| MODIFIER         | NULL         | CHAR(30)    | Last Modifier                                  |
| NEXT_ITEMID      | NOT NULL     | NUMBER(9)   | Next Available Object ID                       |
| OWNER            | NULL         | CHAR(30)    | current owner                                  |
| PAGE_HEIGHT      | NULL         | NUMBER(3)   | Page Height                                    |
| PAGE_WIDTH       | NULL         | NUMBER(3)   | Page Width                                     |
| PARAM_HINT       | NULL         | CHAR(W)     | Hint Line for Parameter Form                   |
| PARAM_STATUS     | NULL         | CHAR(50)    | Status Line for Parameter Form                 |
| PARAM_TITLE      | NULL         | CHAR(80)    | Title for Parameter Form                       |
| REPORT_NAME      | NULL         | CHAR(80)    | Report Name                                    |
| RIGHT_MARGIN     | NULL         | NUMBER(3)   | Right Margin                                   |
| TOP_MARGIN       | NULL         | NUMBER(3)   | Top Margin                                     |
| VERSION          | NULL         | NUMBER(5)   | SQL*ReportWriter Version<br>When Created       |

**SRW\_STE**

| <i>Name</i> | <i>Null?</i> | <i>Type</i> | <i>Comment</i>                                                          |
|-------------|--------------|-------------|-------------------------------------------------------------------------|
| APPID       | NOT NULL     | NUMBER(9)   | Report Id                                                               |
| ITEMID      | NOT NULL     | NUMBER(9)   | Object Id                                                               |
| NAME        | NULL         | CHAR(80)    | object Name                                                             |
| OWNER       | NULL         | CHAR(W)     | Report owner                                                            |
| TYPE        | NULL         | NUMBER(2)   | Object Type:<br>2=Query, 3=Group,<br>4=Field, 5=Summary,<br>6=Parameter |

**SRW\_SUMMARY**

| <i>Name</i> | <i>Null?</i> | <i>Type</i> | <i>Comment</i>                                                                                                     |
|-------------|--------------|-------------|--------------------------------------------------------------------------------------------------------------------|
| APPID       | NOT NULL     | NUMBER(9)   | Report Id                                                                                                          |
| DATATYPE    | NULL         | NUMBER(1)   | summary Datatype:<br>1=NUM, 2=CHAR, 3=DATE                                                                         |
| FIELDID     | NOT NULL     | NUMBER(9)   | Field to Summarize                                                                                                 |
| FORMAT_MASK | NULL         | CHAR(40)    | Display Format                                                                                                     |
| ITEMID      | NOT NULL     | NUMBER(9)   | Summary Id                                                                                                         |
| OPERATOR    | NULL         | NUMBER(2)   | summary operator<br>1=Sum, 2=Min, 3=Max,<br>4=Count, 5=Av~ 6=%Total,<br>+50 for Running Summary,<br>7=First 8=Last |
| OWNER       | NULL         | CHAR(W)     | Report Owner                                                                                                       |
| PRINT_AT    | NULL         | NUMBER(9)   | Print Group                                                                                                        |
| RESET_AT    | NULL         | NUMBER(9)   | Reset Group                                                                                                        |
| SUM_ORDER   | NULL         | NUMBER(3)   | Order on Summary Sreen                                                                                             |
| WIDTH       | NULL         | NUMBER(3)   | summary field Width                                                                                                |

**SRW\_TEXT**

| <i>Name</i>   | <i>Null?</i> | <i>Type</i> | <i>Comment</i>                                                                                                          |
|---------------|--------------|-------------|-------------------------------------------------------------------------------------------------------------------------|
| APPID         | NOT NULL     | NUMBER(9)   | Report Id                                                                                                               |
| DIRTY         | NULL         | CHAR(1)     | Text:<br>X=Edited,Null=Default                                                                                          |
| FREQUENCY     | NULL         | NUMBER(9)   | Frequency                                                                                                               |
| ITEMID        | NOT NULL     | NUMBER(9)   | Object Id<br>1=Report, 2=Page,<br>else Group's Id                                                                       |
| JUSTIFICATION | NULL         | NUMBER(1)   | Justification<br>1=Left, 2=Right,<br>3=Center                                                                           |
| LINES_BEFORE  | NULL         | NUMBER(3)   | Lines Before                                                                                                            |
| OWNER         | NULL         | CHAR(W)     | Report owner                                                                                                            |
| RELATIVE_POS  | NULL         | NUMBER(1)   | Relative Position:<br>1=Below, 2=Right,<br>3=Panel, 4=Margin                                                            |
| REPEAT        | NULL         | CHAR (1)    | Repeat on page overflow                                                                                                 |
| SPACES_BEFORE | NULL         | NUMBER(3)   | Spaces Before                                                                                                           |
| TYPE          | NULL         | NUMBER(2)   | Text Object Type:<br>3=Title Page,<br>4=Trailer Page,<br>5=Header, 6=Footer,<br>7=Column Heading<br>8=Body, 9=Subfooter |
| WIDTH         | NULL         | NUMBER(3)   | Width if Text contains<br>Variable Field                                                                                |

**SRW\_TEXT\_LONG**

| <i>Name</i> | <i>Null?</i> | <i>Type</i> | <i>Comment</i>   |
|-------------|--------------|-------------|------------------|
| APPID       | NOT NULL     | NUMBER(9)   | Report Id        |
| ITEMID      | NOT NULL     | NUMBER(9)   | Text Object Id   |
| OWNER       | NULL         | CHAR(W)     | Report Owner     |
| PANEL       | NULL         | NUMBER(2)   | Panel Number     |
| Text        | NULL         | LONG        | Text of Panel    |
| TYPE        | NULL         | NUMBER(2)   | Text Object Type |

**PRODUCT\_PROFILE** The PRODUCT\_PROFILE table, which is maintained by the DBA, contains restrictions on the user at the product level. Examples of this use are SQL\*Plus's restrictions on commands or SQL\*ReportWriter's per-user page limit. By use of views and a public synonym (PRODUCT\_PRIVS), users can only see those rows in PRODUCT\_PROFILE which apply to them.

| <i>Name</i>   | <i>Type</i>   | <i>Comment</i>           |
|---------------|---------------|--------------------------|
| PRODUCT       | CHAR(W)       | 'SQL*ReportWriter'       |
| USERID        | CHAR(W)       | 'PUBLIC' or userid       |
| ATTRIBUTE     | CHAR (240)    | 'PAGE_LIMIT'             |
| SCOPE         | CHAR (240)    | profile, i.e., French    |
| NUMERIC_VALUE | NUMBER (15,2) | Maximum Number of Pages  |
| CHAR_VALUE    | CHAR (240)    | Character if a Character |
| DATE_VALUE    | DATE          | Date, if a Date          |
| LONG_VALUE    | LONG          | Long String, if a Long   |

**Note:** SQL\*ReportWriter does not use columns CHAR\_VALUE, DATE\_VALUE, and LONG\_VALUE to maintain per-user page limits.

**USER\_PROFILE**

The USER\_PROFILE table is an internal table created for use by SQL\*ReportWriter.

## Additional Objects When System-owned SQL\*ReportWriter Tables Are Installed

### PRODUCT\_ACCESS

The PRODUCT\_ACCESS table is owned by the DBA and used by users to grant access to other users on one or more of their reports. SQL\*ReportWriter has two views of this table: SRW\_GRANT, which is of all grants the user has made, and SRW\_GRANTED, which is of all grants that have been made to the user.

### SRW\_GRANT

(view of the PRODUCT\_ACCESS table)

| <i>Name</i>      | <i>Type</i> | <i>Comment</i>                 |
|------------------|-------------|--------------------------------|
| GRANTEE          | CHAR(240)   | User to Whom Access is Granted |
| NUMERIC_ID APPID | NUMBER (9)  | Report Id                      |
| OWNER            | CHAR(W)     | Owner of Report                |
| PRODUCT          | CHAR(W)     | 'SQL*ReportWriter'             |

### SRW\_GRANTED

(view of the PRODUCT\_ACCESS table)

| <i>Name</i>      | <i>Type</i> | <i>Comment</i>                 |
|------------------|-------------|--------------------------------|
| GRANTEE          | CHAR(240)   | User to whom Access is Granted |
| NUMERIC_ID APPID | NUMBER (9)  | Report Id                      |

### SRW\_NEXT\_APPID

(sequence, used to generate APPID's)

| <i>Name</i> | <i>Type</i> | <i>Comment</i>    |
|-------------|-------------|-------------------|
| NEXT_APPID  | NUMBER(9)   | Next Appid to Use |

## Additional Objects When User-owned SQL\*ReportWriter Tables Are Installed

### SRW\_GRANT

| <i>Name</i>      | <i>Type</i> | <i>Comment</i>                    |
|------------------|-------------|-----------------------------------|
| GRANTEE          | CHAR(240)   | User to Whom Access is<br>Granted |
| NUMERIC_ID APPID | NUMBER (9)  | Report Id                         |
| OWNER            | CHAR(30)    | Owner of Report                   |
| PRODUCT          | CHAR(30)    | 'SQL*ReportWriter'                |

### SRW.GRANTED

(view of the SRW\_GRANT table)

| <i>Name</i>      | <i>Type</i> | <i>Comment</i>                    |
|------------------|-------------|-----------------------------------|
| NUMERIC_ID APPID | NUMBER (9)  | Report Id                         |
| GRANTEE          | CHAR(240)   | User to Whom Access is<br>Granted |

### SRW\_NEXT\_APPID

(sequence of the SRW\_REPORT table if using the ORACLE RDBMS Version 6.0 or later, and SQL\*ReportWriter Version 1.1.12 or later)

(view of the SRW\_REPORT table if using ORACLE RDBMS Version 5, and SQL\*Report Writer Version 1.1.11 or earlier)

| <i>Name</i> | <i>Type</i> | <i>Comment</i>    |
|-------------|-------------|-------------------|
| NEXT.APPID  | NUMBER(9)   | Next Appid to Use |

## APPENDIX

# D

# THE SQL\*REPORTWRITER CALL INTERFACE

The SQL\*ReportWriter Call Interface allows the designer of an application using one of the ORACLE Programmatic Interfaces to make procedure calls to the SQL\*ReportWriter programs SQLREP, RUNREP, GENREP, DUMPREP, and LOADREP. Each procedure *takes* one parameter, a valid SQL\*ReportWriter command line for the program being called. All SQL\*ReportWriter Call Interface calls return a positive number if an error is encountered, or a zero if no error is encountered.

Syntax The five SQL\*ReportWriter procedure calls are (illustrated here in 'C'):

| <i>Command</i> | <i>Procedure</i>               |
|----------------|--------------------------------|
| SQLREP         | long rwcsql ( /*_ char * _*/ ) |
| RUNREP         | long rwcrun ( /*_ char * _*/ ) |
| GENREP         | long rwcgen ( /*_ char * _*/ ) |
| DUMPREP        | long rwcdmp ( /*_ char * _*/ ) |
| LOADREP        | long rwclod ( /*_ char * _*/ ) |

Examples An example of each SQL\*ReportWriter procedure call (in a typical 'C' program) follows

```
long ret, rwcsql () , rwcrun () , rwcgen () , rwcdmp () , rwclod () ;
```

- a) if (ret = rwcsql ("userid=scott/tigerparamform=no"))  
goto error;
- b) if (ret = rwcrun ("report=areportuserid=scott/tigermyparam=AB"))  
goto error;
- c) if (ret = rwcgen ("file=expdat.rexpath=/usr/dsmith"))  
goto error;
- d) if (ret = rwcdmp ("rep-ts=red,repuserid=scottjtiger"))  
goto error;
- e) if (ret = rwclod ("file=larrysreportuserid=scott/tiger"))  
goto error;

Listed below is an example C program that calls the SQL\*ReportWriter procedure RWCRUN. This program was run on VMS using a standard VAX compiler, and linked using the LNSQLREP.COM file located in the same directory.

```
#include <stdio.h>

long ret,
 rwcrun (),
 rwcsql ();

main ()
{
 char options [1321];

 strcpy (opt ions, "userid=scott/tiger report=t batch=yes
paramform=
no");
 strcat (options, " destype=file desname=report .txt");
 printf ("\nOptions are: %s", options);
 ret=rwcrun (options);
 printf ("\nValue of return code is %d", ret);
 ret=rwcrun (options);
 printf ("\nValue of return code is %d", ret);
 exit () ;
}
```

**Rules**

1. If a userid and password is included on the command line, as in the examples above, SQL\*ReportWriter will attempt to log onto that ORACLE account.
2. If the calling program has already logged on to ORACLE, a userid and password should not be included in the command line, so that SQL\*ReportWriter will use the logon that is already active.
3. SQL\*ReportWriter will logoff ORACLE only if it logged on. If the calling program had previously logged on, the logon will still be active on return from SQL\*ReportWriter.
4. If SQL\*ReportWriter logs on, it will COMMIT before logging off (SQLREP or RUNREP), and will neither COMMIT nor ROLLBACK if the calling program had already logged on.
5. Character strings passed to any SQL\*ReportWriter procedure call must be null terminated, meaning that the last valid character of the string must be followed by an ASCII value of zero.

**Calling Procedures  
from Languages Other  
than C**

To call SQL\*ReportWriter procedures from programming languages other than C, you must be aware of how subroutine arguments are passed in your host language: Ada, COBOL, FORTRAN, Pascal, or PL/1.

**Example Programs  
Calling RWCRUN**

Listed below are five example 3GL programs that call the SQL\*ReportWriter procedure RWCRUN. These programs were run on VMS using standard VAX compilers, and linked using the LNSQLREP.COM file located in the same directory.

Note: The exact syntax of your subroutine calls maybe dependent upon your host language compiler. Refer to your compiler documentation for more details on interfacing with C procedure calls.

## Ada

```
with TEXT_IO; use TEXT_IO;
procedure srwada is
 usr : string (1..11) := "Scott/Tiger";
 s : STRING (1..16 O);
 RES : STRING (1..1 O);
 LEN : INTEGER;

 function rwcrun (S : string) return Integer;
 pragma interface (C,rwcrun);
 pragma import_function (internal=RWCRUN,
 PARAMETER_TYPES=(STRING),
 MECHANISM= (REFERENCE),
 RESULT_MECHANISM=VALUE,
 RESULT_TYPE=INTEGER) ;

 package INT_IO is new INTEGER_IO (INTEGER) ;use INT_IO ;
 ret_code : INTEGER;

begin
 s(1..56) := "REPORT=TUSERID=SCOTT/TIGER PARAMFORM = NO
 DESNAME=ADA. TXT";
 S(57..79) := "DESTYPE=FILE BATCH=YES";
 S(80) := standard.ASCII.nul; -- null terminating the string

 PUT_LINE ("About to call rwcrun#1");
 ret_code :=rwcrun (S);

 if ret_code=0 then
 PUT_LINE ("Rwcrun was called successfully ");
 else
 PUT_LINE (" the returned code is"); put (ret_code) ; new_line;
 PUT_LINE (" Error occurred when Runform was called");
 end if;

 PUT_LINE (" About to call rwcrun#2 ");
 ret_code :=rwcrun (S);

 if ret_code=0 then
 PUT_LINE ("Rwcrun was called successfully ");
 else
 PUT_LINE ("the returned code is "); put (ret_code) ;new_line;
 PUT_LINE ("Error occurred when Runform was called");
 end if;

end srwada;
```

```

COBOL IDENTIFICATION DIVISION.
PROGRAM-ID. COBREP.

ENVIRONMENT DIVISION .

DATA DIVISION .
WORKING-STORAGE SECTION.
*
01 OPTIONS.
 02 OPTION-TEXT PIC X (132) VALUE "REPORT=SAMPLE1
USERID=SCOTT/TIGER BATCH=YES PARAMFORM=NO DESTYPE=FILE
DESNAME=SAMPLE1.TXT".
 02 FILLER PICS9(4) COMP VALUE 0.
*
01 RETURN-CODE PICS9(9) COMP.

PROCEDURE DIVISION.
MAIN-PROGRAM .
CALL "RWCRUN "
 USING BY REFERENCE OPTIONS
 GIVING RETURN-CODE .
DISPLAY "Returncode after first call to rwcrun is "RETURN-CODE
CALL "RWCRUN "
 USING BY REFERENCE OPTIONS
 GIVING RETURN-CODE .
DISPLAY "Return code after second call to rwcrun is "RETURN-CODE
STOP RUN:

```

```

FORTRAN program fortran

 integer*4 status
 integer*4 rwcrun

 character*100 command

 command = 'report=test userid=system/manager batch=yes
1 destype=file desname=fortran.txtparamform=no'

 status= rwcrun (%REF(command))
 write(*,l) status
1 format (lx,14)
 status =rwcrun (%REF(command))
 write(*,l) status

 end

```

```

Pascal PROGRAM srwpas (input, output);

Type
 short= [WORD] -32768..32767;
 OptType=packed array[1..256]of char;
 [external] function rwcrun (Var options : OptType) : short;
 Extern;

Var
 Options : OptType;
 RetCode : short;

Begin
 Opt ions := 'USERID=SCOTT/TIGER BATCH=YES REPORT=PASCAL
 PARAMFORM=NO ';
 Options[56] := CHR(0);
 RetCode := rwcrun(options) ;
 Writeln ('Return code from first call is ', RetCode);
 RetCode := rwcrun(options) ;
 Writeln ('Return code from second call is', RetCode);
End.

```

```

PL/1 PLI: PROCEDURE OPTIONS (MAIN);

DECLARE OPTIONS CHAR(100),
 PART1 CHARACTER DEFINED OPTIONS,
 PART2 CHARACTER DEFINED OPTIONS POSITION (51),
 ENDCHAR CHARACTER(1) DEFINED OPTIONS POSITION (100);

DECLARE RWCRUN ENTRY (CHARACTER (100)) RETURNS (FIXED
 B I N A R Y) ;
DECLARE RETCODE FIXED BINARY (15);

PART1='USERID=SCOTT/TIGER BATCH=YES REPORT=T DESNAME=PLI.';
PART2='TXTDESTTYPE=FILEPARAMFORM=NO';
ENDCHAR= LOW (1);
PUT SKIP EDIT ('Options are: ', OPTIONS) (A,A);
RETCODE = RWCRUN (OPTIONS);
PUT SKIP EDIT ('ReturnCode is', RETCODE) (A, B(15));
PUT SKIP (2) LIST ('*****');
RETCODE = RWCRUN (OPTIONS);
PUT SKIP EDIT ('ReturnCode is', RETCODE) (A, B(15));
END PLI;

```

# E

## USER EXITS

This chapter discusses SQL\*ReportWriter user exits. Specifically, it covers the following topics

- what a user exit is
- what types of user exits are available
- what steps are needed to use a user exit in SQL\*ReportWriter
- how to build a user exit
- how to link a user exit
- how to pass arguments to a user exit
- how to return values from a user exit
- how to conditionally highlight fields using user exits
- what restrictions there are on writing user exits
- which user exits are pre-packaged and how they function.

This section on creating a user exit covers an ORACLE Recompiler user exit-the type of user exit you will use most often. Note that some of the details presented in this section will vary with the operating system on which you are working and the language in which you choose to write the user exit.

Read this chapter to get an understanding of user exits and use the appropriate *ORACLE Installation and User's Guide* to guide you in your actual work. Note that before you attempt to implement a user exit, you should know how to compile object files and link executable (SQLREP, RUNREP) on your operating system. Previous exposure to the ORACLE Precompilers is also very useful.

**Note:** In this chapter, any reference to the *ORACLE Precompiler User's Guide* is equivalent to the *Pro\* Precompiler User's Guide*.

## What a User Exit Is

A user exit is a subroutine that you write and link into the SQL\*ReportWriter executables. When you call this subroutine from SQL\*ReportWriter, *control* temporarily passes processing to the user exit. When a user exit has finished executing control is returned to SQL\*ReportWriter.

## Uses for User Exits

User exits can:

- perform complex data manipulation
- compute mathematical functions
- pass data to SQL\*ReportWriter from operating system text files
- manipulate LONG RAW data
- execute PL/SQL blocks and SQL commands.

## Types of User Exits

You can write the following types of user exits:

- ORACLE Recompiler user exits
- OCI (Oracle Call Interface) user exits
- non-Oracle user exits.

You can also write a user exit that combines both the ORACLE Recompiler Interface and the OCI.

## **ORACLE Precompiled User Exits**

An ORACLE Precompiled user exit incorporates the ORACLE Recompiler Interface. This interface allows you to write a subroutine in one of the following host languages and embed SQL commands:

- Ada
- C
- COBOL
- FORTRAN
- Pascal
- PL/I.

With embedded SQL commands an ORACLE Recompiler user exit can access ORACLE databases. Such a user exit can also access SQL\*ReportWriter fields, summaries, and parameters. Because of these features, you might want to write most of your user exits as ORACLE Precompiled user exits.

For more information on the ORACLE Precompiled Interface, refer to the *ORACLE Precompiler User's Guide* for the host language in which you are interested.

Note: Not all operating systems support all of the listed languages. For details, refer to the ORACLE *Installation and User's Guide* for your system.

## **OCI (Oracle Call Interface) User Exits**

An OCI user exit incorporates the Oracle Call Interface. This interface allows you to write a subroutine that has calls to ORACLE databases. A user exit that incorporates only the OCI (and not the ORACLE Precompiled Interface) cannot access SQL\*ReportWriter fields, summaries, and parameters.

For more information on the OCI, refer to a *ORACLE Precompiler User's Guide*.

## **Non-ORACLE User Exits**

A non-ORACLE user exit does not incorporate either the ORACLE Precompiled Interface or the OCI. For example, a non-ORACLE user exit might be written entirely in 'C.'

By definition, a non-ORACLE user exit cannot access ORACLE databases or SQL\*ReportWriter fields, summaries, and parameters.

## Steps Needed for User Exits in SQL\*ReportWriter

We suggest the following steps to produce a report that contains a user exit

- Create your report without user exits
- Decide on which user exits are needed
- Build the function(s) you want in one of the host programming languages
- Fill in the IAPXTB table by calling GENXTB (GENXTB will create a local IAPXTB source file.)
- Compile the IAPXTB source file and the function you created, and then link them to SQLREP and/or RUNREP
- Create the field, or fields, for the user exit(s) and reference the exit in the Source column of Field Screen One
- Enter each user exit name, followed by parameters, that you wish to pass to your function.

## Building a User Exit

This section first explains the basic steps of how to write an ORACLE Precompiled user exit, and then it explains how to pass arguments to that user exit through the Source column of Field Screen One. This section applies specifically to ORACLE Precompiled user exits because it is most flexible, allowing access to both SQL\*ReportWriter objects, such as fields, and to ORACLE database data.

An example of an ORACLE Precompiled user exit follows (in 'c').

```
/' Copyright (c) 1987 by Oracle Corporation */
#include <ctype. h>

/**** ***** */
/* Define user exit routine to copy from one field to another */
/* */
/' Parse assumes that SQL* ReportWriter removes extra spaces */
/* between fields in the #string */
/**** ***** */

/' This is a sample "

#define SQLCA_STORAGE_CLASS extern
EXEC SQL INCLUDE SQLCA;
```

```

/*Error buffer--SQL*ReportWriter will raise error if first byte */
/*non-null */
extern char SRWERB [2561;

int rwecpf (ue_string,len)
char *ue_string;
int *len;
/*len is not used since ue string is always null-terminated */

EXEC SQL BEGIN DECLARE SECTION;
VARCHAR field[240];
VARCHAR value[240];
EXEC SQL END DECLARE SECTION;

char arg [241];
char *p;
char *srw_source;
char *srw_dest;
int i;

/*copy#string to work buffer */
arg [240] = '\0';
strcpy (arg,ue_string) ;

/*skip over user exit name in #lstring*/
p = arg;
while (*p&& !is space(*p)) p++;
if (*p) p++;
else goto error;

/*parse # string for source field; null terminate the source name */
srw_source= p;
while (*p && !isspace(*p)) p++;
if ((p == srw_source) || (!*p)) goto error;
*p++= '\0?;

/* parse # string for destination field; null terminate dest name*/
srw_dest =p;
while (*p && !isspace (*p)) P++;
if (p== srw_dest) goto error;
* P= '\0';

```

```

/* put field value into host variable */
field.len = strlen (strcpy (field.arr, srw_source));
EXEC IAF GET :field INTO :value;

/*put host variable value into field */
field.len=strlen(strcpy(field.arr,srw_dest));
EXEC IAF PUT :field VALUES (:Value);

return(0);

error:
strcpy ((char *) SRWERB, "Usage: #RWECPF <source> <destination>") ;
return(0);
}

```

All ORACLE Precompiler user exits can use host language statements to perform procedural operations. These user exits can also use the following elements to perform additional functions.

| <i>Element</i>          | <i>Use</i>                                          |
|-------------------------|-----------------------------------------------------|
| EXEC SQL statements     | performing SQL commands                             |
| EXEC ORACLE statements  | executing ORACLE Precompiled options                |
| EXEC IAF GET statements | passing values from SQL*ReportWriter to a user exit |
| EXEC IAF PUT statements | passing values from a user exit to SQL*ReportWriter |

## **EXEC SQL Statements**

An EXEC SQL statement is a SQL command, prefixed with "EXEC SQL." EXEC SQL statements allow you to perform any SQL command in an ORACLE Recompiler user exit. It is your means of selecting or manipulating data in the database from a user exit.

**Syntax** EXEC SQL SQL\_Statement;

where:

*SQL\_Statement* is any valid ORACLE SQL statement, except for the restricted commands noted later in this section

Note that you do not need to perform an explicit CONNECT in an ORACLE Precompiled user exit because SQL\*ReportWriter establishes the connection automatically. However, ORACLE does not support concurrent connects. For more information, refer to the appropriate *ORACLE Precompiler User's Guide*.

Restricted Commands Oracle Corporation recommends strongly that you do not issue any command that would implicitly cause a database commit, such as a DDL command, when you are using SQL commands in an ORACLE Precompiled user exit.

See your ORACLE *Installation and User's Guide* to learn which ORACLE Precompiled version is being used for your database.

## **EXEC ORACLE Statements**

An EXEC ORACLE statement is a statement that is not standard SQL and is used to execute ORACLE Precompiler options. For more information, refer to the appropriate *ORACLE Precompiler User's Guide*.

## **EXEC IAF GET**

An EXEC IAF GET statement passes a value from SQL\*ReportWriter to an ORACLE Precompiled user exit. Specifically, it places the value of a SQL\*ReportWriter field, summary, or parameter into a host language variable. Once SQL\*ReportWriter passes a value to a user exit, the user exit can use the value for calculation, manipulation, or update.

**syntax:** EXEC IAF GET report\_variable\_1, report\_variable\_2, . . .  
INTO :host\_variable\_1, :host\_variable\_2, . . . ;

where :

*report\_variable\_n* the name of the SQL\*ReportWriter field, summary, or parameter from which you are reading a value

*host\_variable\_n* the name of the host language variable into which you are reading a SQL\*ReportWriter field, summary, or parameter

Note that *report\_variable\_n* can be a reference to:

- a SQL\*ReportWriter field, summary, or parameter
- a host language variable name (prefixed with a colon) into which you are reading a value.

Refer to the appropriate *ORACLE Precompiler User's Guide* for any restrictions on host language variables.

You must represent host variables and constants in standard SQL format. Dates are passed using the SQL\*ReportWriter format mask of YYYYMMDDHH24MISS. The following table provides examples:

| Value                   | Result                                                                                                                                           |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| :holder1                | inserts the value of the host variable, holder1                                                                                                  |
| 'Summit Sporting Goods' | inserts the constant string value, Summit Sporting Goods. Note that some languages use double quotation marks instead of single quotation marks. |
| '413.60'                | inserts the constant numeric value, 413.60                                                                                                       |
| ' 19880414083000'       | inserts the date value of April 14,1988,8:30 A.M.                                                                                                |

See your ORACLE Installation and User's Guide to learn which ORACLE Precompiled version is being used for your database.)

## EXEC IAF PUT

An EXEC IAF PUT statement passes a value from an ORACLE Precompiled user exit to SQL\*ReportWriter. Specifically, it places the value of a constant or the value of a host language variable into a SQL\*ReportWriter field, summary, or parameter..

syntax EXEC IAF PUT report\_variable . . .  
VALUES ( [ :host\_variable | constant] , . . . ) ;

where

*report\_variable* the name of the SQL\*ReportWriter field, summary, or parameter into which you are reading a value

*host\_variable* the name of the host language variable from which you are reading

*constant* the constant that you are reading (Do not precede a constant with a colon.)

The *report\_variable* can be a reference to:

- a SQL\*ReportWriter field, summary, or parameter
- a host language variable (prefixed with a colon) whose value is any of the above items.

Notes: Refer to the appropriate *ORACLE Precompiler User's Guide* for any restrictions on host language variables.

You must represent host variables and constants in standard SQL format. Dates are passed using the SQL\*ReportWriter format mask of YYYYMMDDHH24MISS.

See your *ORACLE Installation and User's Guide* to learn which ORACLE Precompiled version is being used for *your* database.



## Linking a User Exit

Linking a user exit involves the following steps:

- creating a table named IAPXTB to hold information about each user exit using the GENXTB program
- entering one row per user exit (not per each call of each exit) in the IAPXTB table using a SQL\*Forms form called GENXTB
- generating a source file also called IAPXTB containing an extract of IAPXTB
- compiling the IAPXTB source file
- compiling the function(s) you've written
- linking the object modules with SQLREP and/or RUNREP.

Some details of implementing user exits are specific to each operating system. Refer to the appropriate ORACLE Installation and *User's Guide* for the steps you need to follow.

Note: If you do not have SQL\*Forms, use SQL\*Plus to enter the rows in the IAPXTB table.

### GENXTB and IAPXTB

Before you can implement a user exit (on any operating system), you should have the GENXTB program and GENXTB form. For details on how to make these facilities available, refer to your *ORACLE Installation and User's Guide*. Refer to the appropriate *ORACLE Installation and User's Guide* for details on creating the IAPXTB table.

You must enter one row in the IAPXTB table for every user exit name. (This condition is true for ORACLE Precompiled user exits whether a user exit is in a file that is precompiled and compiled all by itself or with several other user exits.) Also, you should maintain all production user exits for a system in one IAPXTB table.

The table on the following page describes each column in the IAPXTB table:

| <i>Column Name</i> | <i>Description</i>                                                                                                                                                         |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XTB\$XNM           | NAME specifies the name of the user exit. (This is not necessarily the name of the file that contains the user exit.) Note that some program languages are case sensitive. |
| XTB\$XTY           | TYPE specifies the program language in which the user exit is written.                                                                                                     |
| XTB\$REM           | REMARKS specifies a text string that describes the purpose of the user exit. The string can have a maximum length of 80 characters.                                        |
| XTB\$CRE           | CREATION specifies the date on which you created a record in the IAPXTB table. When you commit a record, the GENXTB form automatically enters a value in this column.      |
| XTB\$MOD           | MODIFY specifies the date on which you last modified a record in the IAPXTB table. When you commit a record, the GENXTB form automatically enters a value in this column.  |

Note: You should keep text versions of your user exits in a separate IAPXTB table in case the table is accidentally deleted.

## Passing Arguments to a User Exit

You pass arguments from SQL\*ReportWriter to a user exit by referencing the user exit in the Source column of Field Screen One. First, you must create a field with which to pass the parameters. Next, move to the Source column and add the arguments you wish to pass after the user exit name:

User Exit Call Syntax #userexit text

where:

*userexit* is the pre-packaged user exit name, or the name of your own user exit. The user exit name may be at most 10 characters in length. (On some operating systems the name may be at most 6 characters. Check with your System Administrator.)

*text* can be arguments, printer codes, constants, or any combination thereof, that you wish to pass to the user exit. The string may be at most 240 characters.

SQL\*ReportWriter scans the text string and adds all field and summary names it recognizes to an internal “dependency” list. It then ensures that these fields contain the most recently computed or fetched value before it calls your user exit. Note that SQL\*ReportWriter ignores all quoted text when building its “dependency” list.

Example An example of a valid user exit call is as follows

```
SALARY WORKHOURS COMMISSION VACATION DAYS BONUS
```

Note: If you wish to pass a value back to the SQL\*ReportWriter field, summary, or parameter, you must pass that field, summary, or parameter name, also.

Your user exit is passed as two arguments the first argument is a string that contains everything after the #sign; the second argument is a pointer to the length of the first argument. For example, passing arguments to a user exit named exit name as

```
#exitname argument1 argument2
```

behaves as though the function associated with that user exit were called

```
functionname (' EXITNAME ARGUMENT1 ARGUMENT2' , len)
```

In this example, len is a pointer to an integer that gives the length of the first argument. Your program must parse the string appropriately to extract the arguments.

## Creating an Error Message in a User Exit

A user exit program can return an error to SQL\*ReportWriter by writing into a named 256 character buffer, SRWERB. This buffer is initialized when the report is executed. If, on returning from a user exit, SQL\*ReportWriter finds the buffer non-empty (i.e., the first character in the buff&is not null), then SQL\*ReportWriter will raise an error, using what it finds in the buffer as the error message.

## Returning Values from a User Exit

The field that has the user exit as its Source is not automatically computed. It is up to the user exit to assign a value to the user exit field through use of the IAF PUT call. (See ‘Building a User Exit Function, EXEC IAF PUT.’)

## Conditionally Highlighting Fields

**Passing Printer Control codes from user Exits** To highlight text or fields conditionally, for example, highlighting only “checking accounts” that are less than zero, do the following:

1. Create a field that has a Source of a user exit, and a Data Type of PRT, above the field you wish to highlight. Note: To enter a Data Type of PRT, you must first enter a pound sign (#) followed by a user exit name in the Source column.
2. Create a user exit for the field you just created that examines the value of the field you wish to highlight. The user exit should also set the field you just created with the appropriate printer code (highlight or normal text).

### Example for Fields

| Field        | Source     | Data Type |
|--------------|------------|-----------|
| Bold_Neg_Val | #eval_acct | PRT       |
| Amount       | Amount     | NUM       |

3. In the text object that contains the field you wish to conditionally highlight, enter the reference of the field you created just before the field you want to conditionally highlight. In the example below, when the Amount field is less than 0, &Bold\_Neg\_Val is set to a printer code for bold. Otherwise, it is set to a printer code for normal text.
4. Enter a printer code (for normal text) after the field you wish to conditionally highlight to turn the highlighting feature off. In the example below, &21 is the printer code for normal text.

### Example for Text

&Bold\_Neg\_Val &Amount &21

---

## Restrictions on Writing User Exits

The following restrictions apply to all user exits

- Rules:**
1. User exit names must follow the rules of your operating system and host language. Be aware that these rules might include case sensitivity roles.
  2. Arguments must be separated by spaces only.
  3. Maximum length of the Source column is 240 characters, including the user exit name. Spaces are counted. The user exit name may be at most 10 characters (On some operating systems the limit is 6 characters. Check with your System Administrator). Text between double quotes is treated as a single variable; fields, summaries, and parameters referenced within double quotes will not be added to the user exit field's dependency list. To reference a literal double quote within a user exit, use \". The \" will be deleted before the argument is passed.
  4. If there is more than one space between the variables you pass, the extra spaces will be deleted from the user exit when it is passed to your function program  
  
For example:                   #my function a b c  
becomes                        #my function a b c
  5. It is the responsibility of the user exit, not SQL\*ReportWriter, to parse the user exit string. This string is null-terminated and passed to the user exit function as the first argument. In addition, the length of the string is passed as the second argument.
  6. If a printer code is passed back to SQL\*ReportWriter, the printer code has default width of 0.
  7. Data that is longer than 256 characters cannot be PUT into a SQL\*ReportWriter field. Likewise, if a field's data is longer than 256 characters, the value returned from an EXEC IAF GET is truncated to the left-most 256 characters.

8. Any fields, summaries, or parameters that are referenced in the user exit string, other than the user exit field (in the Field Name field) will be added to a dependency list. Any SQL\*ReportWriter field, summary, or parameter in this list will be computed before the user exit is called. It is therefore recommended that all fields that are referenced in the user exit program be passed as arguments to the user exit. This will guarantee that all fields referenced by the user exit have already been computed. There is no guarantee, however, that a user exit for a field will be calculated on the page on which the field is to be printed.
9. You cannot perform host language screen 1/0 from an ORACLE Recompiler user exit. This restriction exists because the run-time routines that a host language uses perform screen 1/0 conflict with the routines that SQL\*ReportWriter uses to perform its 1/0. You can, however perform host language file 1/0 from a user exit.
10. SQL\*ReportWriter COMMITS changes to the database just after executing a report (with SQLREP or RUNREP). If you wish to COMMIT at any other time, you can do so with your user exit.
11. If you use your own user exits through GENXTB, you must add any pre-packaged user exits you wish to use to the IAPXTB table. (This can also be done through GENXTB.)

## **User Exits Pre-packaged with SQL\*ReportWriter**

There are five user exits packaged with SQL\*ReportWriter that you may use RWECPF, RWEWSC, RWEIF, RWERTH, and RWENOP. Below you will find a brief explanation and the syntax of each one.

### **RWECPF**

This user exit copies the data in source and places the result in destination

**Syntax** #rwecpf source destination

where

*source* the SQL\*ReportWriter field name

*destination* the SQL\*ReportWriter field name.

## RWEGSC

This user exit puts into *destination* the name of the SELECT column defined by query *queryid* and select list position in the report with *appid*. This user exit is designed for use in reports documenting the structure of SQL\*ReportWriter reports.

syntax #rwegsc destination appid queryid position option

where

|                    |                                                                                                 |
|--------------------|-------------------------------------------------------------------------------------------------|
| <i>destination</i> | the SQL*ReportWriter field name to get the SELECT column name                                   |
| <i>appid</i>       | the SQL*ReportWriter field name of the field containing the APPID                               |
| <i>queryid</i>     | the SQL*ReportWriter field name of the field containing the QUERYID                             |
| <i>position</i>    | the SQL*ReportWriter field name of the field containing the position of the column in the query |
| <i>option</i>      | a number (0 or 1).                                                                              |

Rules If *option*= 0, then no query name is prefixed to the column; if *option* =1, then the query name is prefixed to the column name if the column name is not unique within the specified report.

## RWEIF

This user exit is a conditional: if (prml log-op prm2) is TRUE, then the SQL\*ReportWriter field, *target*, receives the value *src1*. If (prml log-op prm2) is FALSE, then the SQL\*ReportWriter field, *target*, receives the value *src2*.

Syntax #rweif prml log-op prm2 target src1 src2

where

|               |                                              |
|---------------|----------------------------------------------|
| <i>prml</i>   | the constant or SQL*ReportWriter field name  |
| log-op        | =, <=, >=, =, !=                             |
| <i>prm2</i>   | the constant or SQL*ReportWriter field name  |
| <i>target</i> | the SQL*ReportWriter field name              |
| <i>src1</i>   | the constant or SQL*ReportWriter field name  |
| <i>src2</i>   | the constant or SQL*ReportWriter field name. |

**Options** DATES: To compare a date to a constant, the constant should be in the form YYYYMMDDHH24MISS.

CONSTANT% Any pair of constants (whether they look like numbers or dates) will be compared character by character. Also, constants must be in double quotes.

**Rules** All constants must be in double quotes.

## **RWERTH**

This user exit removes the text highlighting characters in SQL\*ReportWriter's SRW TEXT LONG.TEXTcchunn from source and places the result in destination. This user exit is designed for use in reports documenting the structure of SQL\*ReportWriter reports.

**Syntax** #rwerth source destination

Where:

*source*                      *the SQL\*ReportWriter field name*  
*destination*                *the SQL\*ReportWriter field name.*

## **RWENOP**

This user exit does nothing but return.

**Syntax** #rwenop

**Options** One case in which you would want to use this user exit is as follows. Say you had the following fields

| <b>Field Name</b> | <b>Source</b> |
|-------------------|---------------|
| A                 | Empno         |
| B                 | #copyAinBC    |
| c                 | #rwenop       |

Where copyAinBC is a user exit that would copy the contents of field A into field B and into field C. If field C did not have #rwenop (a dummy user exit), it would be impossible to copy the contents of A into C.

# F

## VERSION 1.0 TO 1.1 CHANGES

This appendix contains a list of new features and their index entries for users that are familiar with SQL\*ReportWriter. The purpose of this list is to enable users who already know SQL\*ReportWriter Version 1.0 to learn the new functionality of SQL\*ReportWriter Version 1.1 more easily by looking up the indexed term of the new feature, you will not need to read this manual cover to cover to learn about the new functionality. Below, the bolded text indicates the new features. The text in parentheses contains the index term.

**Single Set of Tables:** (See System-owned)

Allows report definitions to be stored in one set of SQL\*ReportWriter tables per database, and provides report-level security. Throughout this manual, this concept is termed “system-owned tables.”

Systemowned tables are only possible when running on an ORACLE database with the transaction processing option. In all other cases, an individual set of “user-owned” SQL\*ReportWriter tables are needed for each user.

**Wrap Text:** (See Wrap)

Allows fields to wrap on word boundaries within their field widths.

**Variable-width Text: (See Variable)**

Allows text and fields to be wrapped on word boundaries within the text object. This allows you to write form letters with correct word-wrapping.

**User Exit Support: (See Source column setting)**

Allows you to integrate 3GL programs into SQL\*ReportWriter. Five user exits are pm-packaged.

**Extended Printer Code Support: (See Highlighting Text)**

Allows you to define custom printer codes to alter the appearance of report outputs, or to embed graphics characters in reports.

**REX to REP conversion without using ORACLE: (See GENREP)**

Allows you to generate runfiles directly from .rex files in the absence of an ORACLE database.

**Call Interface Capability (See Appendix D)**

Allows you to link RUNREP or SQLREP with a user-written program.

**SQL\*Connect Support in RUNREP: (no index term)**

Allows you to use SQL\*Connect to access data in certain non-ORACLE databases.

**Referencing Fields in Queries: (See Query-SQL statement)**

Allows you to reference SQL\*ReportWriter fields and summaries in queries.

**First and Last Summary Functions: (See Summary Screens-Function)**

Allow users to printout the first or last value of a field, or to use the first or last value in a summary computation. This is useful for referencing fields in Page Headers and Footers.

**Page Summaries:** (See Summary Screens-data type)

Allows you to have summaries calculated and printed on page boundaries. This enables users to compute page totals. You can also reference fields and summaries in page headers and footers.

**Lexical Parameters.** (See Parameter Objects-Lexical variable)

Allows you to create both bind and lexical parameters in queries.

**Multi-query Read Consistency without Locking** (See RUNREP-READONLY)

Allows you to retrieve data that is consistent across multiple queries, without locking the tables being queried. Note that this feature is only available when using SQL\*ReportWriter with ORACLE Version 6.0 and later versions.

**Alternate Formats for System Variables:** (See Source Column heading)

Allows you to use &DATE, &PAGE, and &NUM\_PAGES in the Field Settings Screen for formatting purposes.

**Run-time Parameter Form Modification Support:** (See Run-time Parameter Form)

Allows you to modify the Run-time Parameter Form title, hint line, and status line. It also allows users to suppress the appearance of parameters on the Run-time Parameter Form.

**Input and Output Format Masks for Report Parameters:** (See Parameter Objects)

Allows you to control the appearance of parameters when you, or other users, enter parameters on the Run-time Parameter Form and/or when the parameter values appear in the report output.

**Current Date Support:** (See DATE system variable)

Allows users to use the current date as a default parameter value.

**Oracle\*Mail Support:** (See DESTYPE

Allows you to send reports to Oracle\*Mail users. The report will be sent as an attached file.

**Maximum Text Object Defaulting** (See Editing Text Objects-Status indicator)

Allows you to have the text object that you modify be marked as “Edited.” In Version 1.0, editing one text object sometimes caused other objects to become “Edited” as well.

**National Language Support:** (See International-Currency setting, LANGUAGE)

Allows you to specify the currency indicator, thousands separator, and decimal character for formatting numbers and also the appropriate national language spellings of date format elements.

# G

## CREATING TERMINAL DEFINITIONS

**T**his Appendix contains information on how to set up terminal definitions. The first part of this appendix discusses how to use the TERMDEF utility a utility you use to create or modify a terminal definition file. The second part of this appendix contains tables which you will need to refer to while using the TERMDEF utility.

---

## Terminal Definition

SQL\*ReportWriter uses encoded files called terminal/key files to uniquely identify terminals and their corresponding keymappings. For example, to use the default terminal/key file for VT320 terminals, you would enter `sqlrep term=srw_320` when you logon. `SRW_320` is an example of a terminal/key file. It provides the information needed to use SQL\*ReportWriter on the specified terminal type.

Your system already has a number of these files provided. However, you may find that you need a new terminal/key file, perhaps because your organization uses terminals for which a file is not provided. You may even decide to create a new file in order to remap the keyboard for a particular terminal type.

For these reasons, SQL\*ReportWriter provides the `TERMDEF` utility. `TERMDEF` uses a key definition file and a terminal definition file to create a new terminal/key file. The key definition file is a text file containing the keymapping. You can copy an existing file and edit it or create a new file. The terminal definition file contains entries for various terminal types. If you move the terminal definitions (`termdef`) file from the public directory, your terminal definitions will not work.

`TERMDEF` is intended for the following users:

- Users who want to create their own, private keymaps.
- Administrators who want to provide alternative public keymaps.
- Administrators who want to change existing keymaps.
- Administrators whose users run the system on terminals for which SQL\*ReportWriter does not provide a terminal/key file.

## Overview

This section provides an overview of the steps involved in modifying a terminal definition. The specific steps involved are explained later in this appendix.

## Creating a Private Keymapping

Users may want to create their own keymapping in order to override the default settings. Perform the following steps to create a private keymapping.

1. In your local directory, create a new key definition file or copy an existing file and edit the copy.
2. Run `TERMDEF`.
3. Use `term= <name of your terminal/key file>` for `SQLREP` or `RUNREP`.

### Changing an Existing Keymapping

To change the keymapping of an existing terminal/key file, just create a new key definition file or edit the existing one and run TERMDEF. To replace the existing keymapping, overwrite the corresponding key definition file. To create an alternate keymapping, copy the existing file, edit it, and assign it a new name. If this keymapping is to be public, you must place it in the same directory that contains the files that SQL\*ReportWriter provides. Private files must be placed in a local directory.

### Supporting a New Terminal

Complete the following steps if SQL\*ReportWriter does not provide a file for a particular terminal type. Check the terminal definitions file (termdef) to be sure that a terminal definition entry for your terminal does not already exist. If you find an entry that matches your terminal type, you may skip step 1 below.

1. Create a new entry for the terminal in the terminal definitions (termdef.dat) file. You can move the file, or copy it and place it into your local directory. Just make sure that you specify the full path name for termdef.dat.
2. Create a new key definition file in the same directory.
3. Run TERMDEF.

### Running TERMDEF

Each time you run TERMDEF, you specify the name of the key definition file you want to use and the entry that corresponds to your terminal type in the terminal definitions file. TERMDEF then produces the encoded terminal/key file.

Enter the following command to run TERMDEF

Syntax `termdef [TERMINAL=] terminal_name [OUTFILE=] out file  
[INFILE=] infile [USERID=] userid [[TDFILE=] tdfile]`

where:

|                      |                                                                                        |
|----------------------|----------------------------------------------------------------------------------------|
| <i>terminal_name</i> | the second name you entered in the terminal names line                                 |
| <i>outfile</i>       | the location (path) and name of the file in which to place the new terminal definition |
| <i>infile</i>        | the location and name of the key definition file                                       |
| <i>userid</i>        | the ORACLE userid                                                                      |
| <i>tdfile</i>        | the location and name of the terminal description file (the default is termdef.dat).   |

The arguments must be entered in the order specified above.

See "Terminal Capability Codes" later in this appendix for information about terminal capabilities and the terminal names line.

Example `termdef vt220 outfile=my_220 in file=my_220.kdf userid=scott/tiger`

In this example, vt220 is the name that references the entry in the terminal definitions file. my\_220 is the name to be assigned to the resulting terminal/key file. my\_220.kdf is the name of the key definition file.

## Key Definition File

SQL\*ReportWriter furnishes a number of key definition files. You can alter these files or create new ones to suit your users' preferences and terminal types. You can copy an existing file or create one from scratch. Edit the file with your choice of editor. The default key definition files end with the suffix .kdf, but you need not adhere to this convention.

The key definition file contains three columns and the entries are separated by colons(:):

| Function | Character Sequence | Key Assignments |
|----------|--------------------|-----------------|
| UNDO:    | \E[E34-\E[34-      | :F20 F20        |

The Function column contains words that indicate a particular function. For a list of the functions, see "SQL\*ReportWriter Functions" later in this appendix.

The Character Sequence column contains the sequence sent by the keystrokes listed in the Key Assignment column.

Example Pressing F20 twice sends the sequence: \E[E34-\E[34-

which executes Undo in VT200 and VT300 series terminals when you use the default terminal/key files.

The Key Assignment column contains any form of notation you prefer to use in order to indicate the keystrokes that invoke the function.

To add comments to this file, begin each comment line with a semi-colon (;).

## Terminal Definitions File (tenndef)

Each terminal definitions file contains a number of entries listing capabilities of different terminals and the sequences that invoke them. In most cases, you will find that an entry for your terminal type already exists. If you cannot find an entry that matches your terminal, create a new entry in the terminal definitions (termdef) file. To create this entry, you compare the listed "Terminal Capability Codes" (found later in this appendix) with those in your terminal documentation, and record all mutual capabilities.

### A sample file section is shown below:

```
Delays stated in the manual are 2 ms for line operations and 52
milliseconds for screen operations.
dOlv100lv100lpt100lpt-100ldec vt100:\
:is=\017\E (B\ E) O\ E[?31\E [201\ E[41\E[m\E[?6h:\
:cl=\E [;H\E [2 J:ce=\E [K:\
:co#80:li#24:bs:cm=\E[i%d;%dH:nd=\E[C:up=\E[A:do=\J:do=\E [B:\ :ks=\
E=: ke=\E:\
:ru=%oTBS\E [%i%d;%dr\E [%d;1f\E E\E [;r:\
:rd=%oTBS\E [%i%d; %dr\EM\E [; r:\
:hO=\200\E [m:hI=\200\E [4m:h2=\200\E [7m:h3=\200\E [4;7m:\
:h4=\200\E [5m:h5=\200\E [4;5m:h6=\200\E [5;7m:h7=\200\E [4;5;7m:\
:h8=\200\E [lm:h9=\200\E [l;4m:hA=\200\E [l;7m:hB=\200\E [l;4;7m:\hC=\
200\E [l;5m:hD=\200\E [l;4;5m:\
:hE=\200\E [l;5;7m:hF=\200\E [l;4;5;7m:\
:sO=\O17:sI=\O16:\
:gl=\244j\234k\204l\2l4m\25On\236q\2l5t\245u\247v\237w\2l2x:
```

## Description Components

The following list describes the components of an entry in the terminal definitions file.

|                     |                                                                                                                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Comment [optional]  | placed at the beginning or the end of the description. Each line of the comment must begin with a pound (#) sign.                                                                            |
| Terminal Names Line | contains at least three names for the terminal separated by vertical bars( ). End this line with a colon (: ) and a backslash (\).                                                           |
| First Name          | a two-character field containing no blanks You can choose any two characters, but the usual format consists of a letter followed by a digit. This name does not have to be unique (i.e.,do). |
| Second Name         | the mnemonic by which TERMDEF references the terminal. You should assign a short name consisting of letters and digits (i.e., vt3100).                                                       |
| Additional Names -  | One or more additional names, one of which should fully identify the terminal. These names may contain blanks (i.e., DEC vt3100 ).                                                           |

## Terminal Capabilities

The rest of the entry consists of descriptions of the capabilities that SQL\*ReportWriter has in common with the terminal type. You compare the capabilities of the terminal with those of SQL\*ReportWriter and record those they have in common using the appropriate SQL\*ReportWriter codes (i.e., :cl=50\E [2J: ).

This description means that a clear screen takes 50 milliseconds and is invoked with the ASCII ESCAPE character, followed by three characters, [2J.

: c1 is the SQL\*ReportWriter code for “clear screen.”

The different types of codes and capabilities are listed in the sections that follow. See “SQL\*ReportWriter Functions” later in this appendix for a list of specific SQL\*ReportWriter Functions.

## TC [optional]

When placed at the end of a section, it indicates that this terminal has all of the same capabilities as the terminal designated, plus those listed in the Terminal Capabilities for the current terminal.

### Example

Terminal A has capabilities 1,2, and 3.

Terminal B has capabilities 1 and 2.

When creating A's entry, you can designate tc=B at the end of the entry as a shorthand way of indicating that terminal A has all of terminal B's characteristics. In the capabilities *section* of A's entry, you list the capabilities that override B's entry (i.e. capabilities that are different or that B does not have). In this case, you would list capability 3 in the entry for terminal A.

## Entering Terminal Capabilities

To enter the terminal capabilities, compare the capabilities listed in 'Terminal Capability Codes' (found later in this appendix) with those documented in your terminal documentation. Record an entry for each capability they have in common.

Use the following guidelines when recording descriptions

- Each description consists of a two-character code, which is usually followed by a character string.
- Each line, except for the terminal names line and any comment lines, must begin with a colon.
- Each line, except for the last, ends with a backslash to indicate that the next line is a continuation of the same entry.
- All descriptions end with a colon.
- You may include more than one description in a given line.
- To make an entry more readable, you can indent lines. If a description continues onto another line, you can indent the continuation, but be sure that the break occurs between fields.

There are three types of terminal capability codes: Boolean, numeric, and string.

|                |                                                                                                                                                                                                                        |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Boolean</i> | <i>specifies</i> whether a terminal supports a particular feature. For example, <code>:bs:</code> indicates that the terminal supports backspace.                                                                      |
| <i>Numeric</i> | <i>specifies</i> counts and sizes for the various terminal characteristics. The code is followed by a pound sign (#) and an integer. For example, <code>:li#24:</code> indicates that the terminal displays 24 lines.  |
| <i>String</i>  | <i>specifies</i> that a particular capability is available and provides the ASCII character sequence that invokes the function. A string capability code is followed by an equal sign (=) and the invocation sequence. |

Some terminals require a delay between executing a function and receiving the next character from SQL\*ReportWriter. The length of this delay in milliseconds must be given as a decimal integer between the equal sign and the string. For example, `:c1=50\E [2 ]` indicates that a clear screen takes 50 milliseconds on the terminal. After sending the sequence, SQL\*ReportWriter provides enough pad characters to provide the delay.

## Octals

Certain sequences require octal values. The octal value of a character is specified by a backslash followed by three digits. Observe the following points when specifying a character's octal value.

- In some cases, you can substitute abbreviations. The following table contains frequently used characters having two-letter backslash abbreviations.

| <i>Abbreviation</i> | <i>Character</i>            | <i>Octal</i> |
|---------------------|-----------------------------|--------------|
| <code>\E</code>     | ASCII ESCAPE character      | 033          |
| <code>\n</code>     | ASCII linefeed (or newline) | 012          |
| <code>\r</code>     | ASCII carriage return       | 015          |
| <code>\t</code>     | ASCII horizontal tab        | 011          |
| <code>\f</code>     | ASCII formfeed              | 014          |

- When a colon is part of an entry and not just a delimiter you must encode the colon with the octal number 072.
- If an invocation sequence contains more than one part, use ASCII NUL (octal 000) as a delimiter between the parts.
- Encode NULs in a multi-part sequence as octal 200.

## Cursor Movement Capabilities

This section describes the options for the `*move cursor to line X, column Y` terminal description, `"cm."` When these options are present, they must be placed after the sequence that signals cursor motion in a `"cm"` sequence. The cursor movement options are listed and described later in this appendix (see `"Options for Cursor Movement Capabilities"`).

A `"move cursor"` sequence directs a terminal to move its cursor to a specific line and column, both of which must be indicated in the sequence. Terminal manufacturers have a variety of mechanisms for encoding this, so the `"cm"` terminal definition entry has to be capable of describing many different formats. Some terminals require that the `"move cursor"` sequence give the line first, then the column; others require the opposite. Some terminals number lines and columns beginning with zero, others begin with one.

The `"cm"` sequence options are marked with percent signs (%) followed by one or more letters that specify the option. The line and column numbers are substituted into the `"cm"` string in place of the option on output. For more details, see a UNIX manual, section 5, on `termcap`.

## Highlighting Codes

SQL\*ReportWriter uses four types of highlighting underlining reverse video, blinking, and altered intensity. There are sixteen combinations of these four types along with the type of highlighting they produce. See "Highlight Codes" later in this appendix.

The code h0 turns all highlighting off. Each highlighting sequence must turn on all highlighting styles it represents and turnoff all the others.

A sequence that invokes highlighting must describe whether a terminal handles highlighting as permanent or transient or both. To do this, the sequence may contain permanent or transient parts, or both. Permanent highlighting marks a physical area on the terminal. Transient highlighting sets a mode that affects only the text displayed, not the area in which it is written.

The first part of a highlighting sequence is not sent to the terminal. Instead, it is a flag byte that signals the system how a terminal handles highlighting. It tells the system whether the sequence for the highlighting code will consist of permanent or transient highlighting parts and where these parts will occur in the invocation sequence that follows.

Permanent highlighting puts a permanent-on mark in the space at the beginning of an area of the terminal. The area that is highlighted extends to the space where the system puts a no-highlighting mark. If new text overwrites text in a permanent area, the new text will be highlighted. Permanent highlighting requires none or one space before and after the highlighted area to mark it; the contents of the space will disappear.

Transient highlighting sets the terminal into highlighting mode. Text sent to the terminal between the time highlighting is turned on and the time it is turned off will be highlighted. If new text overwrites transiently-highlighted text, the new text is not highlighted. Transient highlighting sets no permanent marks, and thus requires no spaces before or after the text.

The value of the flag byte is the sum of the octal values of the sequence's parts, followed by a number of spaces (0-3) that are required to mark the beginning and end of an area of permanent highlighting. The octal value of the parts of a highlighting sequence are:

|        |               |
|--------|---------------|
| Ž \200 | Transient-On  |
| Ž \100 | Transient-Off |
| Ž \040 | Permanent-On  |

The value in the flag byte reflects the presence of one to three of the parts. They must appear in the sequence in the order they are listed above if more than one is present. Each part of a sequence that precedes another part must end in \000. For example, in the entry "h8=\341\E)\000\E(\000\EG0:" the first \000 indicates the end of the transient-on part and that the beginning of the transient-off part follows. If a sequence must contain an ASCII NUL (\000), encode it as \200.

For example, a highlighting sequence containing only a transient-on part would have a flag byte of octal 200. A sequence containing a transient-on part, a transient-off part, and a permanent-on part requiring one space to set would have an octal value of 200+100+40+1, or 341.

#### Initialization Sequence

The initialization sequence contains the characters that SQL\*ReportWriter sends to a terminal when it begins executing. The sequence must set up the terminal screen and keyboard for cursor movement and other editing functions. It should also turn off unused features such as automatic margins and initial highlighting.

See "Terminal Capability Codes" later in this appendix for the SQL\*ReportWriter codes and other information you may need. The Sample Terminal Definitions File Entry section takes you through each part of a sample terminal type file.

#### Sample Terminal Definitions File Entry

# Delays stated in the manual are 2 ms for line operations and 52 # ms for screen operations.

```
d0lvltl001vt-100lpt100lpt-100ldecvt100:\
:is=\017\E (B\ E) 0\ E[?31\E [201\ E[41\E[m\E [?6h\E[?11:\
:c 1 = 5 2 \ E [; H \ E [2 J : c e = \ E [K : \
:co#80:li#24:bs:cm=\E[%i%d;%dH:nd=\E[C:up=\E[A:do="J:do=\E[B:\
:ks=\E=:ke=\E>:\
:ru=%oTBS\E[%i%d;%dr\E[%d;lf\EE\E[;r:\
:rd=%oTBS\E[%i%d;%dr\EM\E[;r:\
:h0=\200\E[m; hl=\200\E [4m:h2=\200\E [7m:h3=\200\E [4;7m:\
:h4=\200\E[5m:h5=\200\E[4;5m:h6=\200\E [5;7m:h7=\200\E [4;5;7m:\
:h8=\200\E[lm:h9=\200\E[l;4m:hA=\200\E[l;7m:hB=\200\E[l;4;7m:\
:hC=\200\E[l;5m:hD=\200\E[l;4;5m:\
:hE=\200\E[l;5;7m:hF=\200\E[l;4;5;7m:\
:s0=\017:s1=\016:\
:gl=\244j\234k\2041\214m\250n\236q\215t\245u\ 247v\237w\2l2x:
```

# Delay stated in the manual are 2 ms for line operations and 52  
# ms for screen operations.

The pound sign (#) at the beginning of each line indicates that this is a comment.

```
d0 I vt100 lvt-100l pt100 I pt-100 ldecvt100:\
```

This is the Terminal Names line.

d0 is the two-character name. vt 100 is the name that SQL\*ReportWriter will use to reference this file. vt -100, pt 100, and pt -100 are alternate names. dec vt 100 is the full name.

```
:is=\017\E(B\E)0\E[?3J\E[201\E[41\E[m\E[?6h:\
```

This is the initialization string, an example of a string type code. It indicates the characters that SQL\*ReportWriter sends to a terminal when it begins executing.

```
:co#80:li#24:
```

This line describes the screen display and cursor movement. :li and :co are examples of numeric type codes.

: co refers to the number of columns displayed. 80 indicates the number of columns displayed. # follows : co because :co is a numeric code. : li refers to the number of lines displayed. 24 indicates the number of lines displayed on the terminal screen.

```
:bs
```

:bs is an example of a Boolean type code. It simply indicates that the terminal supports a backspace (AH).

The lines that contain codes ho to h9 and hA to hF are highlighting codes. The flag byte \200 that begins each sequence indicates that all highlighting is Transient-On.

## Tables and Lists

### Highlight Codes

| <i>code</i> | <i>Type</i>   | <i>Underline</i> | <i>Reverse Video</i> | <i>Blink</i> | <i>Altered Intensity</i> |
|-------------|---------------|------------------|----------------------|--------------|--------------------------|
| h0          | string        | off              | off                  | off          | off                      |
| h1          | string        | ON               | off                  | off          | off                      |
| h2          | string        | off              | ON                   | off          | off                      |
| h3          | string        | ON               | ON                   | off          | off                      |
| h4          | string        | off              | off                  | ON           | off                      |
| h5          | string        | ON               | off                  | ON           | off                      |
| h6          | string        | off              | ON                   | ON           | off                      |
| h7          | string        | ON               | ON                   | ON           | off                      |
| h8          | string        | off              | off                  | off          | ON                       |
| h9          | string        | ON               | off                  | off          | ON                       |
| hA          | <i>string</i> | off              | ON                   | off          | ON                       |
| hB          | string        | ON               | ON                   | off          | ON                       |
| hC          | string        | off              | off                  | ON           | ON                       |
| hD          | string        | ON               | off                  | ON           | ON                       |
| hE          | string        | off              | ON                   | ON           | ON                       |
| hF          | string        | ON               | ON                   | ON           | ON                       |

### Terminal Capability Codes

| <i>Codie</i> | <i>Type</i> | <i>Description</i>                                         |
|--------------|-------------|------------------------------------------------------------|
| al           | string      | Insert a blank line before the cursor line.                |
| bc           | string      | Move the cursor one space left (backspace) if not with ^H. |
| bs           | Boolean     | Move the cursor one space left (backspace) with 'H.        |
| ce           | string      | Clear from the cursor position to the end of the line.     |
| cl           | string      | Clear the entire terminal display.                         |
| cm           | string      | Move the cursor to a specific line and column.             |
| co           | number      | Number of columns displayed on a terminal line.            |
| dc           | string      | Delete one character at the cursor position.               |
| dl           | string      | Delete the line that the cursor is on.                     |
| do           | string      | Move the cursor down one line.                             |
| ds           | string      | Initialize the down-loadable character set.                |
| ei           | string      | End insert mode and enter overtype mode.                   |

**Terminal Capability Codes (continued)**

| <i>code</i>    | <i>Type</i> | <i>Description</i>                                                                                         |
|----------------|-------------|------------------------------------------------------------------------------------------------------------|
| g0-g3          | string      | Defines the line-drawing characters in the given character set (see "s0-s3 and g0-g3" below).              |
| ic             | string      | Insert a character at then cursor position, shifting other text to the right.                              |
| im             | string      | Enter insert mode.                                                                                         |
| is             | string      | Initialize the terminal for any number of terminal options (e.g., cursor movement and screen editing.      |
| ke             | string      | Exit special "keypad transmit" mode.                                                                       |
| ks             | string      | Enter "keypad transmit" mode, where the numeric keypad sends special command sequences instead of numbers. |
| li             | number      | Number of lines displayed on the terminal.                                                                 |
| nd             | string      | Move the cursor to the right without overwriting text.                                                     |
| p <sub>c</sub> | string      | Pad characters for delays (if not ASCII NUL \000).                                                         |
| qs             | Boolean     | Query the terminal state for restoring on SQL*ReportWriter termination (for VT220s only.)                  |
| rd             | string      | Downward scroll in a region of the display.                                                                |
| rl             | string      | Left scroll in a region of the display.                                                                    |
| rr             | string      | Right scroll in a region of the display.                                                                   |
| ru             | string      | Upward scroll in a region of the display.                                                                  |
| s0-s3          | string      | Defines the given character set (see "s0-s3 and g0-g3" below).                                             |
| tc             | string      | Cross-reference to another terminal that shares capabilities not listed in this terminal's description.    |
| te             | string      | Terminate cursor-motion mode.                                                                              |
| ti             | string      | Initialize cursor-motion mode.                                                                             |
| up             | string      | Move the cursor up one line.                                                                               |

## s0-s3 and g0-g3

SQL\*ReportWriter supports up to four different character sets using s0 through s3 and g0 through g3. The first character set (s0/g0) must contain all of the printable characters. All four character sets can contain any number of line drawing characters.

s0 through s3 are the strings that start the given character set. g0 through g3 are strings containing two-byte pairs that identify the line drawing characters available in the given character set. The first byte of the pair indicates the line drawing character (see the table below). The second character of the pair is the byte to send to the terminal to get the desired line drawing character.

| <i>Line Drawing Character</i> | <i>First Byte of Pair</i>                                               |
|-------------------------------|-------------------------------------------------------------------------|
| up arrow                      | A                                                                       |
| down arrow                    | B                                                                       |
| 5-line horizontal             | C                                                                       |
| others                        | $128 + 27*\text{left} + 9*\text{above} + 3*\text{right} + \text{below}$ |

Where left, above, right, and below are the number of line segments (0, 1, or 2) left, above, right, or below the center of the character cell.

Examples 'T-bar' has one segment left, right, and below the center. The first byte is  $128+27+3+ 1 = 159 = 227$  octal.

"=" has two segments left and two segments right of center. The first byte is  $128+ 27*2 + 3*2 = 188 = 274$  octal.

In the sample Terminal Definitions File Entry, shown earlier in this appendix, the first two bytes of g1 are \244j. They mean the following. send a "j" in character set s1 to get a ( $244 = 164 = 128+27+9 =$  one segment left + one segment above) backward L (the lower right corner of a box).

## Options for Cursor Movement Capabilities

| <i>Option</i> | <i>Description</i>                                                                                                                                                                                                                                                                                                                             |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| %r            | Reverse the line and column output order-send column first and then line. By default, the system substitutes the line number followed by the column number.                                                                                                                                                                                    |
| %i            | Increment the line and column numbers. By default, the system starts lines and columns at 0; with this option, the system starts lines and columns at 1.                                                                                                                                                                                       |
| %>ab          | If either the line or the column number is greater than the octal value of the number in the "a" position, add to that number the octal value of the number in the "b" position. In the example %>\101 \050, line 5 would be left as is and column twelve would be converted to 52 (octal 050 is decimal 40, plus 12 is decimal 52).           |
| %n            | Exclusive-Or the line and column numbers with octal 0140.                                                                                                                                                                                                                                                                                      |
| %0            | Determine the order in which the coordinates are output. %0 requires the next three characters to be T, B, and S, which indicate the following the top, bottom, and the number of lines to scroll. %0 can only be used with :ru and :rd. See the "Sample Terminal Definitions File Entry" earlier in this appendix for an example of using %0. |
| %B            | Convert the line and column numbers to BCD before displaying them (add to each number 6 times that number divided by 10) to display as one character.                                                                                                                                                                                          |
| %D            | For each line and column number: divide by 16, multiply the remainder by 2, and subtract the result from the line or column number. This is required by the Delta Data terminal.                                                                                                                                                               |
| %%            | Display a single percent sign (does not affect when the line or column is displayed).                                                                                                                                                                                                                                                          |

The remaining options must be placed in the sequence after any of the preceding options.

|    |                                                                                                                                  |
|----|----------------------------------------------------------------------------------------------------------------------------------|
| %d | Display the line or column number as an integer. For example, the system would send column 25 as the two characters "2" and "5". |
| %2 | Same as %d, but display the number as two digits. For example, the system would display 2 as 02.                                 |
| %3 | Same as %d, but display the number as 3 digits. For example the system would display 6 as 006.                                   |

- `%+n` Add the octal value of the number in the “n” position to the number and display the result as one character. `%+A` would display 5 as ASCII F (5 + A= F).
- `%.` Display the number as the ASCII character to which its octal value is equivalent. The system would display column 39 as a single quote(') since ASCII 39 equals octal 047.

## **SQL\*ReportWriter Functions**

The SQL\*ReprtWriter functions are listed and described in Appendix A.

Terminal definitions that are consistent with SQL\*Forms 3.0 and SQL\*Menu 5.0 have been added to SQL\*ReportWriter 1.1. See the key definition files `srw_100c.kdf` and `srw_220c.kdf` and their derived terminal definition files (`vt100`, `vt220`, `vt320`, and `vt340`) in the terminals subdirectory.

# *H*

## PRINTER DRIVERS

This appendix contains instructions for defining your printer drivers, including information about:

- the default drivers provided with SQL\*ReportWriter
- creating and modifying the printer definition file
- running the PRIINTDEF command.

## Printer Driver Definitions

The PRINTDEF utility allows you to create drivers for other printers or to create new versions of the default drivers. The PRINTDEF command translates a human-readable definition file into a compressed format file for use by SQLREP and RUNREP. To create a new driver you must:

- create a definition file
- run the PRINTDEF command referencing the definition.

## Default Drivers

Some examples of drivers that SQL\*ReportWriter provides are

|          |                                                                                                               |
|----------|---------------------------------------------------------------------------------------------------------------|
| dflt     | a generic file that ignores highlighting attributes and supports 66x80 page sizes.                            |
| wide     | a generic file that ignores highlighting attributes and supports 66x132 page sizes.                           |
| wide180  | same as wide, but supports 66x180 page sizes.                                                                 |
| dec      | a generic file for most DEC printers; supports 66x80 page sizes for the LN03, LPS40, LP05, and LA50 printers. |
| decwide  | same as dec, but supports 66x132 page sizes.                                                                  |
| decland  | a generic file that prints in landscape mode and supports 66x132 page sizes.                                  |
| dec180   | same as decland, but supports 66x180 page sizes.                                                              |
| hpl      | a generic file for the HP LaserJet printer; supports 66x80 page sizes.                                        |
| hplwide  | same as hpl, but supports 66x132 page sizes.                                                                  |
| pspor80  | a generic file for PostScript printers; supports 66x80 page sizes.                                            |
| pspor132 | same as pspor80, but supports 66x132 page sizes.                                                              |
| pslan80  | a generic file for PostScript printers that prints in landscape mode; supports 40x80 page sizes.              |
| pslan132 | same as pslan80, but supports 60x132 page sizes.                                                              |
| pslan180 | same as pslan80, but supports 66x180 page sizes.                                                              |

## Rule When Using Drivers

When using the driver definitions, make sure that the report height setting for your report matches the :li value in your driver definition. If they do not match, SQL\*ReportWriter will use both values when formatting the report, causing extra form feeds in the report output.

## Creating/Modifying the Printer Definition File

The definitions for the default drivers are stored in two files, called *printdef.dat* and *pstscript.dat*. See your *Installation and User's Guide* to learn the location of these files. (Most installations will allow you to copy these files to serve as a starting point for creating new definitions.)

A single definition file may contain definitions for one or more printers. The definition for each printer contains two parts a label and a set of formatting codes. The structure of each part is described below.

Label syntax printer id | printer name | comment: \  
where

*printer id* used for internal purposes. The printer id must be exactly two letters and unique within the file.

*printer name* the label which references the driver description in the PRINTDEF command. It must be between 1 and 128 characters long.

*comment* used for documenting the capabilities of the driver.

For example:

```
DC | DEC1 | DEC printer, includes 66x80 output:\
```

Formatting Code Syntax format code [=string | #number]

where

*format code* one of the codes listed in the table below, or a number for a user-defined printer code. No spaces are allowed between codes, strings, and numbers.

*string* one or more characters that maybe preceded by an equal sign (=). Control characters are preceded by an up arrow ( ^ ), and escape sequences are preceded by \. Multiple sequences are allowed in one string.

If no equal sign and string is specified after the format code, then no character sequence is used for that format code in the report.

*number* always preceded by a pound sign (#).

Example      04=^X^Y  
               ff=^L  
               so=\E[1m  
               li#66  
               04=\E[ON

Example Definition      The following is an example of a complete driver definition:

```
DC IDEC11 DEC printerj includes 66x80 output:\
:ff=^L:cr="M:nl="J:is= ^L:so=\E [lm:se=\E [Om:li#66:co#80\
:us=\E[4m:ue=\E [Om:ap: 56=\E[2N
```

| PrinterFormatCodes | Name | Type    | Description                                                                                                                                                                                                                      |
|--------------------|------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | am   | Boolean | Determines whether to set automatic margins e.g.,addaCR/LF if there is a character in the 80th column of a report that has a page width of 80                                                                                    |
|                    | ap   | Boolean | Does not generate a form feed if the printer page height (in the printer definition) is equal to the report page height (on the Report screen). This code is important when the top and bottom margins of a report are set to O. |
|                    | bc   | string  | Backspace character string                                                                                                                                                                                                       |
|                    | bs   | Boolean | Determines whether to allow backspacing                                                                                                                                                                                          |
|                    | co   | number  | Number of columns in a line                                                                                                                                                                                                      |
|                    | Cr   | string  | Carriage return (default=^M)                                                                                                                                                                                                     |
|                    | ff   | string  | Printer page eject (default=^L). To create reports with no formfeeds, specify :ff with value (e.g., :ff=:cr=^M)                                                                                                                  |
|                    | fs   | string  | Printer termination string                                                                                                                                                                                                       |
|                    | is   | string  | Printer initialization string                                                                                                                                                                                                    |
|                    | li   | number  | Number of lines on a page                                                                                                                                                                                                        |
|                    | d    | string  | Newline string                                                                                                                                                                                                                   |
|                    | nnn  | string  | 'nnn' is any number signifying a user-defined printer control code. Any string or escape character can be issued                                                                                                                 |
|                    | os   | Boolean | Determines whether to set overstrike, e.g., print a character, backspace, reprint the character                                                                                                                                  |
|                    | Ps   | Boolean | Determines whether the printer is a PostScript printer                                                                                                                                                                           |
|                    | se   | string  | End highlight mode                                                                                                                                                                                                               |
|                    | so   | string  | Begin highlight mode                                                                                                                                                                                                             |
|                    | ue   | string  | End underscore mode                                                                                                                                                                                                              |
|                    | us   | string  | Begin underscore mode                                                                                                                                                                                                            |

- Rules
1. A colon(:) must begin the list of codes.
  2. A colon must end the list of codes.
  3. A colon must separate each code definition.
  4. Format codes must appear in lower case.
  5. Format code descriptions cannot be broken across multiple lines.
  6. A \ at the end of each string associated with a format code acts as a continuation character.
  7. A # at the beginning of a line signifies a comment line. There is no limit to the number of comment lines.
  8. Boolean codes take effect if the format code is present (e.g., :am, :ps). Otherwise, the capability is not enabled.
  9. Lines cannot exceed 512 characters (or the operating system limit).

## Running the PRINTDEF Command

After putting the driver definition in the `printdef.dat` or `pstscript.dat` file, run the following example PRINTDEF command to compile the driver.

Syntax `printdef [PRINTER=] printer [OUTFILE=] out file [ PDFFILE=]pdfile]`

where:

|                |                                                                                                    |
|----------------|----------------------------------------------------------------------------------------------------|
| <i>PRINTER</i> | the name of the printer (taken from the second part of a printer definition label)                 |
| <i>OUTFILE</i> | the location (path) and name of the file in which to place the compiled printer definition         |
| <i>PDFFILE</i> | the location and name of the printer description file (the default is <code>printdef.dat</code> ). |

Example The following are examples of using the PRINTDEF command:

```
printdef dec decbold
```

```
printdef pspor80 pshilite pstscript .dat
```

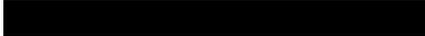
# I

## ERROR MESSAGES LISTING

**T**his appendix lists the errors that may occur before the SQL\*ReportWriter error file has been opened, and then lists all error messages for the SQL\*ReportWriter utilities:

- SQLREP
- RUNREP
- GENREP
- DUMPREP
- LOADREP
- PRINTDEF.

**Note:** “Abnormal condition” errors are not listed in this appendix. Refer to the last page in this appendix for information on what you should do if you receive such an error.



## Start-up Error Messages

When an error is raised and the error message cannot be obtained from the error file, then SQL\*ReportWriter will display one of the following messages

- Error raised but unable to open error file.
- Still attempting to report error.

Then, one of the following hard-coded error messages will appear:

- He not found: <filename>
- Error opening file <filename>
- ORACLE error number <ORACLE error number>  
<ORACLE description of the error>
- Bad argument in command list. Argument was argument>
- No runfile was specified.
- Error opening specified terminal description file <filename>
- Unable to get message for error <error number>  
<Additional information. <information>>

There is one other time a hard-coded error message will appear. When an error occurs and for some reason the error subsystem does not execute correctly, then the following will appear:

- The error manager is unable to report an error due to recursive failure of the error subsystem. Aborting.

## Utility Error Messages

### **1000: Another object already exists with this name.**

**Cause:** Likely cause: there already is a query, group, field, summary, or parameter with this name.

Other possible causes: the report itself already has this name, or you used a reserved word (i.e., REPORT or PAGE).

**Action:** Enter a name that is unique in the report you are currently defining.

### **1001: There is no group with this name.**

**Cause:** The name you entered does not identify any group in the report you are currently defining. You may have mistyped the name.

**Action:** Press [List] and select one of the groups.

### **1002: There is no query with this name.**

**Cause:** The name you entered does not identify any query in the report you are currently defining. You may have mistyped the name.

**Action:** Press [List] and select one of the queries.

### **1003: There is no field with this name.**

**Cause:** The name you entered does not identify any field in the report you are currently defining. You may have mistyped the name.

**Action:** Press [List] and select one of the fields.

### **1004: There is no object with this name.**

**Cause:** The name you entered does not identify any object in the report you are currently defining. You may have mistyped the name.

**Action:** Press [List] and select one of the objects.

**1005: There is more than one source column with this name.**

**Cause:** A column with this name appears in more than one query in the report you are currently defining. You may have mistyped the name.

**Action:** Press [List] and select one of the columns.

SQL\*ReportWriter distinguishes between columns with identical names by adding a prefix to the column name. The prefix identifies the query where the column appears.

**1006: There is no source column with this name.**

**Cause:** None of the queries in your report has a column with this name. You may have mistyped it.

**Action:** Press [List] and select one of the columns.

If you do not see the column you are looking for, check your SELECT Statements on the Query screen. You may have to add this column to one of the queries.

**1007: There is no source column with that name in this query.**

**Cause:** The SELECT Statement that you defined on the current screen does not include a column with the name that you entered. You may have mistyped the name.

**Action:** Press [List] and select one of the columns.

If you do not see the column you are looking for, add it to the SELECT Statement on the current screen.

If there are any aliases in your SELECT Statement, the corresponding field name is the alias, not the column specification.

**1008: There is no source column with that name in this query's parent.**

**Cause:** The parent query you identified does not have a column with the name that you entered. You may have mistyped the name.

**Action:** Press [List] and select one of the columns.

If the one you want does not appear, add it to the SELECT Statement of the parent query.

**1009: You must specify a print direction for this group.**

**Cause:** A group must have a Print Direction; this setting cannot be blank.

**Action:** Press [List] and select an option for Print Direction.

**1010: A query cannot be its own parent.**

**Cause:** You entered the name of the query that you defined on the current screen.

**Action:** Press [List] and select one of the queries.

**1012: Queries must be named.**

**Cause:** You tried to leave Query Name without entering a name for the current query.

**Action:** Enter a name for the query you are defining. Query names must conform to SQL naming conventions.

If you do not want to define anew query, press [Delete Record]. Each report must have at least one query.

**1013: Groups must be named.**

**Cause:** You tried to leave Group Name without naming the group that you are currently defining.

**Action:** If you want to define a new group, enter a name that conforms to SQL naming conventions.

If you do not want to define anew group, press [Delete Record].

**1014: Fields must be named.**

**Cause:** You tried to leave Field Name without naming the field that you are currently defining.

**Action:** If you want to define a new field, enter a name that conforms to SQL naming conventions.

If you do not want to define a new field, press [Delete Record].

**1015: Summaries must be named.**

**Cause:** You tried to leave Summary Name without naming the summary that you are currently defining.

**Action:** If you want to define anew summary, enter a name that conforms to SQL naming conventions.

If you do not want to define anew field, press [Delete Record].

**1016: You must enter a SELECT Statement.**

**Cause:** There are two situations that cause this message

1. You named a query, but tried to leave the current screen without entering SELECT Statement.

2. You named a query and entered a SELECT Statement, but you either specified a statement that does not begin with the keyword SELECT or you did not enter any statement at all.

**Action:** Enter a SQL statement that begins with the word SELECT (you may use upper or lower case). No other SQL statements are allowed.

**1017: The object you have referenced is not a query.**

**Cause:** You tried to leave a setting which expects a query name, but the name you entered is not a query (it is a group, a field, a summary, or a parameter).

**Action:** Press [List] and select one of the queries.

**1018: The object you have referenced is not a group.**

**Cause:** You tried to leave a setting which expects a group name, but the name you entered is not a group (it is a query, a field, a summary, or a parameter).

**Action:** Press [List] and select one of the groups.

**1019: The object you have referenced is not afield.**

**Cause:** You tried to leave a setting which expects a field name, but the name you entered is not a field (it is a query, a group, a summary, or a parameter).

**Action:** Press [List] and select one of the fields.

**1020: You have not specified a parent for this query.**

**Cause:** You tried to leave Parent 1 Columns after entering a value, but Parent Query 1 is blank.

**Action:** Press [Delete Record] to remove the value from Parent 1 Columns (the SELECT Statement on this screen will not be deleted; each pair of matching columns makes up a single detail record and you can specify multiple pairs).

If you want to match columns, first move the cursor to Parent Query 1 and press [List]. Select a query, then enter 1 or more pairs of matching columns.

**1021: The value for Lines Before can be blank or between 0 and 999.**

**Cause:** You entered a value less than 0 or greater than 999.

**Action:** Enter a value between 0 and 999, inclusive, or leave this attribute blank.

**1022: The value for Spaces Before can be blank between 0 and 999.**

**Cause:** You entered a value less than 0 or greater than 999.

**Action:** Enter a value between 0 and 999, inclusive, or leave this attribute blank.

**1023: This query cannot be the Parent Query because it is a descendant of the current query.**

**Cause:** The query you entered is a descendent (child, grandchild, etc) of the query that you defined on the current screen.

**Action:** Press [List] and select one of the queries.

**1024: This object cannot have text associated with it.**

**Cause:** The name you entered references a query, a field, a summary, or a parameter. However, a text object can only be associated with a group, a page, or a report.

**Action:** Press [List] and select one of the groups, or REPORT or PAGE.

**1025: You can not summarize date, page number, number of pages, or a printer code.**

**Cause:** You tried to define a computed field for a field that has a Source of &DATE, &PAGE, &NUM\_PAGES, or PRT.

**Action:** Define a summary (instead of a computed field) for fields with those Sources. Create a new field on the Summary Settings Screen, press [List] and select the field you wish to summarize (the one with the Source of &DATE, &PAGE, &NUM\_PAGES, or PRT), specify a Function, and a Print and Reset Group.

**1026: Valid options are 'Below', 'Right', 'Panel' or blank.**

**Cause:** You entered a value for Relative Position on the Field Screen that is not valid.

**Action:** Press [List] and select an option for Relative Position, or leave this attribute blank to accept the default value.

**1027: Invalid Text Type for this object See the List of values.**

**Cause:** You entered a text Type that is invalid for the item that you referenced in Object.

**Action:** Press [List], and select one of the text Types listed. Object. You can also press Fetch without specifying anything for text Type.

**1028: This name does not conform to SQL naming standards.**

**Cause:** The name you entered violates one or more SQL naming conventions.

**Action:** Enter a name that conforms to these conventions.

**1029: This SELECT statement has duplicate column names.**

**Cause:** SQL\*ReportWriter does not allow the same column name to appear in the SELECT list more than once.

**Action** If you want to select a column multiple times, use aliases to uniquely identify the extra references to this column.

If you only want to select each column once, make sure your statement makes single references to all columns.

**1030: You must enter a matching column for this query.**

**Cause:** Either you tried to specify a column for the parent without specifying one for the child, or you tried to leave the child blank.

**Action:** Press [List], and select one of the columns shown.

If you do not want to match columns, press [Delete Record], and SQL\*ReportWriter will only delete the matching column information. The SELECT Statement on this screen will not be deleted; each pair of matching columns makes up a single detail record, and [Delete Record] only removes a single detail or master record at a time.

**1031: You must enter a matching column for the parent query.**

**Cause:** You identified a matching column for the child query (the one that is defined on the current screen) but you have not yet identified which column you want to match it to in the parent query.

**Action:** Move the cursor to Parent 1 Columns (or Parent 2 Columns, if applicable). Then press [List], and select one of the columns listed.

If you do not want to match columns, press [Delete Record], and SQL\*ReportWriter will only delete the matching column information. The SELECT Statement on this screen will not be deleted; each pair of matching columns makes up a single detail record, and [Delete Record] only removes a single detail or master record at a time.

**1032: Valid options are 'Always', 'Conditional' or blank.**

**Cause:** You entered an invalid option for Page Break, then tried to leave this attribute.

**Action:** Press [List] and select an option for Page Break or leave this attribute blank and the page will break whenever it is full.

**1033: You cannot rename a system parameter.**

**Cause:** You tried to change the Parameter Name of a system parameter on the Parameter Screen.

**Action** To create a new parameter on the Parameter Screen, press [Insert Record Above], or [Insert Record Below] to create a new record, enter a parameter name and its corresponding attributes, assign a default value (if desired), and use the parameter in a SELECT Statement (if desired).

**1034: Your Top Margin plus Bottom Margin is greater than the Page Height.**

**Cause:** The value in Top added to the value in Bottom exceeds the value in Height. That is, the top and bottom margins together are greater than the length of the page.

**Action:** Make sure the value in Height equals or exceeds the sum of Top and Bottom.

**1035: Your Left Margin plus Right Margin is greater than the Page Width.**

**Cause:** The value in Left added to the value in Right exceeds the value in Width. That is, the left and right margins together are greater than the width of the page.

**Action:** Make sure the value in Width equals or exceeds the sum of Left and Right.

**1036: The value for Page Height must be between 1 and 999.**

**Cause:** You entered a value less than 1 or greater than 999.

**Action:** Enter a value between 1 and 999, inclusive.

**1037: The value for Page Width must be between 1 and 999.**

**Cause:** You entered a value less than 1 or greater than 999.

**Action:** Enter a value between 1 and 999, inclusive.

**1038: The value for Right Margin must be between 0 and 999.**

**Cause:** You entered a value less than 0 or greater than 999.

**Action:** Enter a value between 0 and 999, inclusive.

**1039: The value for Bottom Margin must be between 0 and 999.**

**Cause:** You entered a value less than 0 or greater than 999.

**Action:** Enter a value between 0 and 999, inclusive.

**1040: Where is another report with this name.**

**Cause:** The ORACLE account that you are logged onto already has a report with the name that you entered. Report names must be unique.

**Action:** Choose another name; make sure you conform to SQL naming standards.

You can also rename the report that already has this name by using the Rename action.

**1041: You must be the owner of the report to perform this operation.**

**Cause:** You either tried to delete a report that you do not own, or you tried to create, rename, or copy a report that is owned by another user.

**Action:** Give your report a different name. If you want to copy a report that is owned by another user, see the rules on the next screen. (Press [Scroll Right].)

If both you, and the creator of the report, are storing report definitions in centralized tables, that user must have granted you access (on the Report Setting Screen) to the report you want to copy before you may copy it.

If either you, or the creator of the report, are storing report definitions in local tables, that user must have granted you access to the report by running the Public (all users) or to a specific person. To grant access to two specific people, the report creator must run the script twice, once for each person.

**1042: You must enter the name of a report**

**Cause:** You cannot perform this action until you enter the name of an existing report.

**Action:** Press [List] and select one of the reports.

**1043: There is no report with this name.**

**Cause:** The name you entered does not identify any report in your SQL\*ReportWriter tables.

**Action:** Press [List] and select one of the reports.

**1044: You cannot use a summary function on a field whose source is a user exit.**

**Cause:** You tried to enter a Compute Function for a field that has a user exit as its Source. Fields with a Source of a user exit can only be computed on the Summary Screen.

**Action:** If you wish to compute a value on the field that has a Source of a user exit, delete the text in the Function column, leave the remaining information for the field as it is on the Field Screen move to the Summary Screen, create a summary with a Field that contains the Field Name of the user exit. Enter the Function you desire.

If do not wish to compute a value for that field, delete the text that is in the Function column.

**1045: You must enter a new name for the report.**

**Cause:** You tried to rename or copy an existing report, but you did not identify the name that you want to use.

**Action:** Enter a name that conforms to SQL naming standards.

Make sure the name is different than the names of existing reports, as well as objects (queries, groups, fields, summaries, or parameters) in the report that you are copying or renaming.

**1046: Reports must be named.**

**Cause:** You tried to perform the New Report action, but you did not specify the name of the new report.

**Action:** Enter a name that conforms to SQL naming standards.

Make sure this name is different than the names of existing reports. If you want to know the names of previously defined reports, press [List], then press [Undo] to move the cursor back to the New Report Window.

**1047: You must associate each group with a query.**

**Cause:** You tried to leave Query, but you have not yet identified which query the group belongs to.

**Action:** Press [List] and select one of the queries.

If you do not want an additional group, press [Delete Record].

**1048: A 'Crosstab' group must be a Matrix Group.**

**Cause:** When the Print Direction for a group is CROSSTAB the group must be a matrix group.

**Action:** To make this a matrix group, enter X in Matrix Group.

If you do not want the group to be a matrix group, first place an X in Matrix Group, then press [Previous Field], and change the value to another valid Print Direction. Then press [Next Field], and delete the X.

**1049: You can not set 'Page Break' for a Matrix Group.**

**Cause:** A matrix group cannot have a Page Break option.

**Action:** Make sure Page Break is blank before leaving this attribute.

**1050: You can not compute Percent, Sum or Average of a character or date field.**

**Cause:** The Data Type for the field you are defining is CHAR or DATE. You cannot use numeric functions (such as Percent, Sum, or Average) unless the datatype is NUM.

**Action:** Press [List], and select a function.

**1051: A report must have three matrix groups or no matrix groups. If there are three matrix groups, the Print Directions must be 'Down', 'Across', and 'crosstab'.**

**Cause:** You attempted to leave the Group Screen with a partial matrix report defined.

**Action:** If any group has an X in Matrix Group, then the report must have exactly 3 groups (each from a different query), and all 3 groups must have an X in Matrix Group. One group must have DOWN for Print Direction, another group must have ACROSS and the third group must have CROSSTAB. See Matrix Group for more information about creating a matrix report.

If you do not want this to be a matrix report, remove the Xs from Matrix Group for all groups. Also make sure the Print Direction for all groups is either DOWN (and/or DOWN/ACROSS) or ACROSS (and/or ACROSS/DOWN).

**1052: 'AcrossDown' or 'Down/Across' groups cannot be part of a matrix report.**

**Cause:** You tried to enter X for Matrix Group, but the Print Direction for the group you are defining is either ACROSS/DOWN or DOWN/ACROSS. These Print Directions are not allowed for a matrix report.

**Action:** If you want this to be a matrix *report*, press Previous Field, and change the Print Direction to either DOWN, ACROSS or CROSSTAB. See Matrix Group for more information about creating a matrix report.

If you do not want this to be a matrix report, remove the X from Matrix Group.

**1053: You are referencing an object which has been deleted.**

**Cause:** The cursor is on a record that has one or more attributes marked \*undefined\*. This occurs when the attribute references an object that has either been deleted or is currently unavailable.

**Action:** There are several different approaches you could take. If the referenced object is unavailable see if you can make it accessible. (Example: fix an inoperative database link to make an inaccessible table available). You can also move the cursor to each attribute(s) that is marked \*undefined\* and reference valid object(s). Or, you could press [Delete Record] and remove the current record. Alternatively, you could press [Undo], and recreate the "undefined" object on another screen.

**1054: Invalid Print Direction. See the List of values.**

**Cause:** You entered an invalid option for Print Direction, then tried to leave this attribute.

**Action:** Press [List] and select an option for Print Direction.

This attribute cannot be blank.

**1055: Valid options are 'Right', 'Below' or blank.**

**Cause:** You entered an invalid option for Relative Position on the Group Screen.

**Action:** Press [List] and select an option for Relative Position, or leave this attribute blank to accept the default.

**1056: Invalid Highlight style. See the List of values.**

**Cause:** You entered an invalid option for Highlight, then tried to leave this attribute.

**Action:** press [List] and select a Highlight style, or leave this attribute blank to accept the default.

**1057: Valid options are 'Left', 'Above' or blank.**

**Cause:** You entered an invalid option for Label Position, then tried to leave this attribute.

**Action:** Press [List] and select an option for Label Position, or leave this attribute blank to accept the default (which depends on the Print Direction).

**1058: The value for Record Spacing can be blank or between 0 and 999.**

**Cause:** You entered a value less than 0 or greater than 999.

**Action:** Enter a value between 0 and 999, inclusive, or leave this attribute blank

**1059: The value for Field Spacing can be blank or between 0 and 999.**

**Cause:** You entered a value less than 0 or greater than 999.

**Action:** Enter a value between 0 and 999, inclusive, or leave this attribute blank

**1060: The value for Fields Across can be blank or between 1 and 999.**

**Cause:** You entered a value less than 1 or greater than 999.

**Action:** Enter a value between 1 and 999, inclusive, or leave this attribute blank.

**1061: The value for Width must be between 1 and 999.**

**Cause:** You entered a value less than 1 or greater than 999.

**Action:** Enter a value between 1 and 999, inclusive.

**1062: You must enter a Source column for this field.**

**Cause:** You tried to leave Source Column without entering the name of a column from one of queries in this report.

**Action:** If you want to define this field, press [List] and select one of the columns.

If you do not want to define this field, press [Delete Record].

**1063: You must specify which group this field is in.**

**Cause:** You tried to leave Group without entering the name of a valid group in this report.

**Action:** If you want to define this field, press [List] and select one of the fields.

If you do not want to define this field, press [Delete Record].

**1064: Invalid group for this Source column.**

**Cause:** You entered the name of a group that comes from a different path than the group the field belongs to. You then tried to leave this attribute.

**Action:** Press [List] and select one of the groups. All the groups listed belong to the correct path.

**1065: A running summary must reset above its print group.**

**Cause:** You entered the name of a Reset Group that is at the same level or lower than the group entered in Print Group.

**Action:** Press [List] and select an option for Reset Group. All the groups listed are higher than the current Print Group.

**1066: The Panel Number must be between 1 and 10.**

**Cause:** You entered a value less than 1 or greater than 10.

**Action:** Enter a value between 1 and 10, inclusive.

If you do not want this panel in your report, press [Delete Record].

**1067: You must specify which panel this text is to be displayed on.**

**Cause:** Panel Number is currently blank, and you tried to leave this attribute.

**Action:** Enter a value between 1 and 10, inclusive.

If you do not want this panel in your report, press [Delete Record].

**1068: The Data Dictionary view of CATALOG must be available to see the List of values.**

**Cause:** You pressed [List] while in the SELECT Statement area, but SQL\*ReportWriter is unable to get the list of available tables because it cannot access the view named CATALOG.

**Action:** Ask your Database Administrator for assistance.

**1069: Invalid Relative Position. See the List of values.**

**Cause:** You entered an invalid option for Relative Position on the Text Screen.

**Action:** Press [List] and select an option for Relative Position, or leave this attribute blank to accept the default.

**1070: Valid options are 'Left', 'Right', 'Center' or blank.**

**Cause:** You entered an invalid option for Justification, then tried to leave this attribute.

**Action:** Press [List] and select an option for Align, or leave this attribute blank to accept the default (which is LEFT).

**1071: Invalid Text Type. See the List of values.**

**Cause:** You entered an invalid option for Text Type, then tried to leave this attribute.

**Action:** Press [List] and select one of the Text Types.

**1072: Invalid Function. See the List of values.**

**Cause:** You entered an invalid option for Function, then tried to leave this attribute.

**Action:** Press [List] and select one of the Functions.

**1073: You must specify a field for this summary to operate on.**

**Cause:** There currently is no value in Field, and you tried to leave this attribute.

**Action:** Press [List] and select one of the summaries.

If you do not want to define this summary, press [Delete Record].

**1074: You must specify a function for this summary.**

**Cause:** There currently is no value in Function, and you tried to leave this attribute.

**Action:** Press [List] and select one of the Functions.

If you do not want to define this summary, press [Delete Record].

**1075: A non-running summary must reset at or above its print group.**

**Cause:** The group you chose for Reset Group is lower than the group currently in Print Group.

It does not make sense for the accumulator to reset more often than it prints, so lower groups are not allowed here.

**Action:** Press [List] and select an option for Reset GRoup. All the groups in this list are the same level or higher than the current Print Group.

**1076: You must specify a print group for this summary.**

**Cause:** There currently is no value in Print Group, and you tied to leave this attribute.

**Action:** Press [List] and select a Print Group.

If you do not want to define this summary, press [Delete Record].

**1077: Summaries can only print at groups above, equal to or below the group the field is from.**

**Cause:** The group you chose for Reset Group is not from the same path as the group that the field being summarized belongs to.

**Action:** Press [List] and select a Reset GROUP. All the groups in this list belong to the relevant path.

**1078: You can not specify a reset group for this field because it is not a computed value field.**

**Cause:** You entered a value in Reset Group without specifying a Function for this field.

**Action:** Make Reset Group blank.

If you want to define a computed field, press Previous Field, then press [List] and select a function. You can now override the default Reset Group, if you would like, by pressing [List] and selecting the name of a valid group.

**1079: A running computed value must reset above the group where it prints.**

**Cause:** The group you chose for Reset Group is at the same level or lower than the group of the field being computed.

It does not make sense for the accumulator of a running computation to reset more often than it prints, so lower groups are not allowed here.

**Action:** Press [List] and a Reset Group. All the groups listed here are higher than the group of the field being computed.

**1080: A non-running computed value must reset at or above the group where it prints.**

**Cause:** The group you chose for Reset Group is lower than the group of the field being computed.

It does not make sense for the accumulator of a computed field to reset more often than it is printed, so lower groups are not allowed here.

**Action:** Press [List] and select a Reset Group. All the groups listed here at the same level or higher than the group of the field being computed.

**1081: Running summaries may not print at REPORT.**

**Cause:** You chose REPORT for the Print Group of the summary that you are defining.

It does not make sense to have a running summary only print once, so REPORT is not allowed here.

**Action:** Press [List] and select a Print Group.

**1082: Invalid Alignment. See the List of values.**

**Cause:** You entered an invalid option for Align, then tried to leave this attribute.

**Action:** Press [List] and select an option for field alignment, or leave this attribute blank to accept the default (which depends on the Data Type).

**1083: This SELECT statement has a parameter with a name that does not conform to SQL naming standards.**

**Cause:** You entered a parameter name that begins with a numeric digit.

**Action:** Change the name of the parameter so that it does not begin with a numeric digit and conforms to SQL naming conventions.

SQL\*ReportWriter interprets any word in a SELECT Statement beginning with a colon(:) to be a parameter.

**1084: The group's Column Heading cannot appear here because this group is not above it in the group hierarchy.**

**Cause:** The group you entered for Frequency is at the same level or lower than the group the current text Object belongs to.

It does not make sense for a Column Heading to print as often or more often than each record in the group, so groups at the same level or lower are not allowed here.

**Action:** Press [List] and select an option for Frequency. All the groups in this list are higher than the group the current text Object belongs to.

**1085: Summaries of fields from 'Down' or 'Across' matrix groups can not print at 'Across' or 'Down' matrix groups, respectively.**

**Cause:** You chose a Print Group that is the DOWN Matrix Group, even though the field being summarized belongs to the ACROSS Matrix Group, or vice versa.

**Action:** Press [List] and select a Print Group. All the groups in this list are places where the summary can print.

**1086: Character data can not have a Display Format.**

**Cause:** You entered a value in Display Format for a field or summary whose Data Type is CHAR.

**Action:** Make sure Display Format is blank for this character field.

**1087: A summary must reset at or above the summarized field's group.**

**Cause:** The group you chose for Reset Group is lower than the group of the field being summarized.

It does not make sense for the summary's accumulator to reset more often than it is printed, so lower groups are not allowed here.

**Action:** Press [List] and select a Reset Group. All the groups listed here at the same level or higher than the group of the field being summarized.

**1088: A computed value must reset at or above the highest group in which the field's Source column is printed.**

**Cause:** The group you chose for Reset Group is lower than the highest group where a field appears that has the same Source Column as the field you are currently defining.

It does not make sense for the accumulator of a computed field to reset more often than it is printed, so lower groups are not allowed here.

**Action:** Press [List] and select a Reset Group. All the groups listed are high enough to meet the criterion described above.

**1089: Warning One of your SELECT statements is invalid because the database has been modified.**

**Cause:** Since the last time you worked on this report either someone modified the tables that appear in your SELECT statement, or there is a problem accessing these tables.

**Action:** Check your SELECT statements and make sure they are all valid. You may have to restore database links, check new table definitions, etc.

**1090: The value for Top Margin must be between 0 and 999.**

**Cause:** You entered a value less than 0 or greater than 999.

**Action:** Enter a value between 0 and 999, inclusive.

**109: The value for Left Margin must be between 0 and 999.**

**Cause:** You entered a value less than 0 or greater than 999.

**Action:** Enter a value between 0 and 999, inclusive.

**1092: You have nothing to undelete.**

**Cause:** You pressed [Undelete Record]. However, there currently is nothing in the buffer to undelete.

If you delete a record, then undelete it, you cannot undelete it a subsequent time (to make another copy of it). You also cannot undelete any object if you executed or generated the report after the deletion.

**Action:** Manually enter the value that you were trying to undelete.

You cannot undelete a deleted query. Instead, SQL\*ReportWriter orders queries according to the query structure of your report. If the report includes sibling queries, you can reorder their corresponding groups.

**1093: In a matrix report, a field cannot appear in both the 'Down' group and the 'Across' group.**

**Cause:** You created two fields with the same Source column, and one field belongs to the ACROSS group, and the other field belongs to the DOWN group.

**Action:** Reference both fields in the same group, or delete one of the fields from your report definition.

**1094: This SELECT statement has a parameter with the same name as another object.**

**Cause:** A parameter referenced in this SELECT Statement has the same name as a query, group, field, summary, or another parameter in this report.

**Action:** Check each parameter in this SELECT Statement and rename the one that duplicates another object.

If you would like to use this name for the parameter, you could rename the other object, then reference the parameter by the desired name.

**1095: You must undelete the system parameter you deleted before continuing.**

**Cause:** You deleted a parameter that is defined in a SELECT statement. You cannot remove this parameter from the Parameter Screen.

**Action:** To continue, press [Undelete Record] or [Undo].

If you would still like to remove this parameter, go to the Query Screen, and delete it from the SELECT statements where it appears.

**1096: Valid options are 'CHAR', 'NUM' or 'DATE'.**

**Cause:** You entered an invalid option for Data Type or deleted the default value for this parameter.

**Action:** Press [List] and select a Data Type. This attribute cannot be blank.

**1097: valid Destination Type. See the List of values.**

**Cause:** You entered an invalid value for DESTYPE on the Parameter Screen or the Run-time Parameter Form.

**Action:** Valid options are Screen, File, Printer, Mail, or Sysout. Sysout can only be used when running reports via RUNREP with the BATCH=YES argument.

**1098: A Query's two parents cannot be the same.**

**Cause:** Both Parent 1 Query and Parent 2 Query currently reference the same query.

**Action:** Press [List] and select a query name that is different than the query you chose for the other parent.

**1099: A Query's second parent cannot be a parent or child of its first parent.**

**Cause:** There can be no parent-child relationships between Parent Query 1 and Parent Query 2.

**Action:** If you are defining a matrix report, decide which of the 3 queries will be the child query. Check each of the other 2 queries (that is, the parent queries), and delete any values in the Parent-Child Relationships area. Then return to the child query, and enter the name of one parent query in Parent Query 1, and name the other parent query in Parent Query 2.

If you are not defining a matrix report, do not enter any value in Parent Query 2.

**1100: A Query cannot have a second parent if it has no first parent.**

**Cause:** You entered a value in Parent Query 2 before entering a value in Parent Query 1.

**Action:** If you are defining a matrix report, make Parent Query 2 blank, then move to Parent Query 1. Press [List] and select one of queries. Press [Next Field] and enter the name of the other parent query.

If you are not defining a matrix report, do not enter a value in Parent Query 2.

**1101: You cannot match a Child Column to columns from both parents on the same line.**

**Cause:** You entered the name of a column in Child Columns, Parent 1 Columns and Parent 2 Columns. This error message appeared when you tried to leave Parent 2 Columns.

**Action:** Make Parent 2 Columns blank. Then follow one of these options:

Option 1: If you are defining a matrix report, put the column(s) that match the child query to the second parent on subsequent line(s). First press [Next Record] and enter a column for Child Columns and another column for Parent 2 Columns.

Option 2: If you are not defining a matrix report, do not enter any value for Parent Query 2.

**1102: You have not specified a second parent for this query.**

**Cause:** You tried to leave Parent 2 Columns after entering a value, but Parent Query 2 is blank.

**Action:** Make Parent 2 Columns blank. Then follow one of these options

Option 1: If you are defining a matrix report, move to Parent Query 2, and enter the name of the second parent query. Then match the child and parent columns accordingly.

Option 2: If you are not defining a matrix report, do not enter any value for Parent Query 2.

**1103: You cannot set 'Multi-Panel' for a Matrix Group.**

**Cause:** You put an X in the Multi-Panel entry field (on Field Screen Three) for a group that has an X in the Matrix Group entry field (on Field Screen One). SQL\*ReportWriter will create extra pages depending on the amount of data queried.

**Action:** If you are creating a matrix report, delete the X in the Multi-Panel entry field. If you are not creating a matrix report, delete the X in the Matrix Group entry field.

**1104: Another user is currently editing this report.**

**Cause:** Another user is currently using the report that you wanted to use.

**Action:** Try calling up the report later.

**1105: Parameters must be named.**

**Cause:** You tried to leave Parameter Name before naming or renaming this parameter.

**Action:** Provide a name that conforms to SQL naming standards. Make sure this name does not duplicate any query, group, field, summary or any other parameter in this report (also make sure it is different than the name of the report).

If you do not want to define a new parameter, press [Delete Record].

**1106: Another person is already using SQL\*ReportWriter while logged on the same ORACLE account as you.**

**Cause:** Another person logged onto the same ORACLE as the one you are currently accessing and this person is using SQL\*ReportWriter.

**Action:** Use another ORACLE account, or wait for this other person to log off.

**1107: Your query structure will not support a matrix report**

**Cause:** The queries in this report do not follow the conventions for defining a matrix report.

**Action:** For information about how to build a matrix report, see the examples in this Help System.

**1108: Your Default Value is longer than the parameter's Width.**

**Cause:** The value that appears in Default Value has more characters than the value in Width.

**Action:** Decrease the width of the Default Value or increase the Width.

**1109: Valid options are 'CHAR', 'NUM', 'DATE' or 'PRT'.**

**Cause:** You entered a Data Type for a field that was not 'CHAR', 'NUM', 'DATE' or 'PRT'.

**Action:** Press [List] and select an option for Data Type or leave this attribute blank to accept the default.

**1110: Unable to access the SQL\*ReportWriter V1.1 tables. They must be installed in your account, or you must be granted access to them, before using SQLREP.**

**Cause:** You have logged onto an ORACLE account that does not have access to a set of SQL\*ReportWriter Version 1.1 tables.

**Action:** Use an ORACLE account that already has access to SQL\*ReportWriter Version 1.1 tables, or install these tables in the account that you are currently using. If you are not familiar with this procedure, ask your SQL\*ReportWriter Administrator for assistance.

**1111: Warning: This report definition has been edited using a later version of SQL\*ReportWriter. Use of SQL\*ReportWriter in this manner is not recommended.**

**Cause:** You opened a report that was used edited in a later version of SQL\*ReportWriter than the version you are using.

**Action:** Upgrade your SQL\*ReportWriter version to the version that was used to edit the report you tried to open. (Note: SQL\*ReportWriter reports are upward-but not downward-comparable.)

**1112: A value other than YES or NO was specified for the command line argument: PARAMFORM.**

**Cause:** You specified "PARAMFORM = value" where the value was neither YES nor NO.

**Action:** Specify YES or NO, or do not use the PARAMFORM parameter.

**1113: You do not have access to the system's PRODUCT.PROFILE table.**

**Cause:** SQL\*ReportWriter has not been properly installed.

**Action:** The PRODUCT\_PROFILE table must be installed in the SYSTEM account by the database administrator before using SQL\*ReportWriter. For details, see the Installation and User's Guide for your environment.

**1114A parameter cannot have the same name as a command line keyword.**

**Cause:** You entered a parameter name in a SELECT Statement or on the Parameter Screen that is the same as a SQL\*ReportWriter command line keyword.

**Action:** Choose a name for this parameter that conforms to SQL naming standards, but is different than the command line reserved words.

**1115: Valid options are 'Screen', 'File', 'Printer' or 'Mail'.**

**Cause:** You entered a value for Destination Type on the Parameter Form that is invalid, or you left this attribute blank.

**Action:** Enter one of the valid options mentioned in the error message.

**1116: You have specified an invalid Printer Description File.**

**Cause:** There is no printer driver with the name that you used. You may have mistyped this name.

**Action:** Enter the name a valid printer driver for (Printer option) Format on the Report Screen. You can also leave this option blank, and SQL\*ReportWriter will use the default Printer Description File called 'dflt'.

If you do not know the names of available drivers, check with your SQL\*ReportWriter administrator or consult the Oracle Installation and User Guide for your operating environment.

**1117: 'Sysout' is valid only when using RUNREP with the command line argument BATCH=YES. To view the report use 'Screen'**

**Cause:** You entered 'Sysout' for the DESTYPE on either the Parameter Screen or the Run-time Parameter Form, but Sysout can only be used when running reports via RUNREP with the BATCH=YES argument.

**Action:** Enter SCREEN, FILE, PRINTER, or MAIL for the DESTYPE. If you want to use Sysout, run the report using RUNREP with the BATCH=YES argument.

**1118:Unable to locate a user exit specified in the report.**

**Cause:** You have a user exit (a Source specified on Field Screen One that begins with #) that SQL\*ReportWriter cannot locate. This is probably because the user exit was not linked properly.

**Action:** Check and make sure that the IAPXTB source file and the function you created were compiled and then properly linked to SQL\*ReportWriter. Then run the report again. .

If you did not mean to create a user exit, delete the pound sign (#) from the Source column on the Field Screen One.

**1119:The value for the Thousands separator or Decimal indicator may only be one character wide.**

**Cause:** You tried to enter more than one character for the THOUSANDS or DECIMAL system parameters on the Parameter Screen or on the Run-time Parameter Form.

**Action:** Enter any value that is at most one character wide.

**1120: You can not change the datatype of a system parameter.**

**Cause:** You attempted to type into the Data Type field on the Parameter Screen for one of the system parameters: DESTYPE, DESNAME, DESFORMAT, COPIES, CURRENCY, THOUSANDS, or DECIMAL.

**Action:** The Data Type of these values is set by SQL\*ReportWriter and it is not meaningful to change them.

**1121: Warning Your attempt to set the READONLY parameter is being ignored. READONLY is for use with RUNREP only.**

**Cause** On the command line, you entered READONLY=YES or YES where the keyword was interpreted to be READONLY. READONLY can only be used with RUNREP.

**Action:** To use READONLY, first create or modify the report using SQLREP and then run the report using RUNREP with READONLY=YES .

**1122: This SELECT statement contains an INTO clause. You cannot use INTO clauses in SQL\*ReportWriter queries.**

**Cause:** You entered a SELECT statement with an INTO clause. You do not need INTO clauses to select values into fields.

**Action:** Remove the INTO clause from your SELECT statement.

**1511: Not enough memory available.**

**Cause:** SQL\*ReportWriter cannot access sufficient memory from the system to complete this task.

**Action** Make more memory available, if you can, or contact the system administrator for this machine.

**1524 The columns in query in err msg> are not currently available.**

**Cause:** Your query defines a SELECT Statement that uses proper SQL syntax, but it references columns that no longer exist in the database. The columns existed when SQL\*ReportWriter initially validated the SELECT Statement, but they are no longer accessible.

**Action:** Check your database tables (by pressing [List]), and make sure the table(s) and/or view(s) referenced in the FROM clause are still available. If they are, make sure the column definitions are still the same. You can also modify the SELECT Statement, if appropriate.

**1532: You cannot execute this report because it does not have any queries, groups or fields.**

**Cause:** You named a report, but you have not yet defined any queries. As a result, there are no objects (including groups and fields) to process.

**Action:** Acknowledge the error message by pressing [Spacebar]. Then move to the Query Screen by pressing [Q], or by moving the highlight to \*'Query" and selecting this item. Enter a name for a query, press [Next Field], then enter a valid SQL SELECT Statement.

You can now execute the report by pressing [Menu], [A] (for "Action"), [E] (for "Execute"), and [Select]. By default, the output will appear on your screen.

**1534: Text <in err msg> exceeds 10 panels (the maximum allowed).**

**Cause** If SQL\*ReportWriter produced the report as it is currently defined, it would have to create more than ten panels. Ten, however, is the maximum allowed in any report.

**Action** There are many ways to reduce the number of panels, such as decreasing the width fields and summaries on the Field Screen and Summary Screen, deleting panels or moving fields closer together on the Text Screen, etc.

Consider the field identified in the error message when estimating how close you currently are to the 10 panel maximum. Estimate how close this field is to the end of the report.

**1535: Text <in err msg> exceeds 10 pages (the maximum allowed).**

**Cause:** If SQL\*ReportWriter produced the report as it is currently defined, it would have to create more than ten panels. Ten, however, is the maximum allowed in any report.

**Action:** There are many ways to reduce the number of panels, such as decreasing the width fields and summaries on the Field Screen and Summary Screen, deleting panels or moving fields closer together on the Text Screen, etc.

Consider the field identified in the error message when estimating how close you currently are to the 10 panel maximum. Estimate how close this field is to the end of the report.

**1542: ORACLE error occurred during logoff.**

**Cause:** You invoked a SQL\*ReportWriter executable (DUMPREP, LOADREP, etc.) and an ORACLE error occurred when SQL\*ReportWriter attempted to logoff of ORACLE.

**Action:** Rerun the executable. If this error occurs repeatedly, contact Oracle Customer Support for assistance.

**1551: The report named <in err msg> already exists in SQL\*ReportWriter's tables.**

**Cause:** You attempted to load into the SQL\*ReportWriter tables a report that already exists there.

**Action:** Rename or drop the report stored in the SQL\*ReportWriter tables, then use LOADREP to load the report (stored in a .rex file) into the tables.

**1554: Not enough memory available for ORACLE to process a SQL\*ReportWriter query.**

**Cause:** ORACLE cannot access sufficient memory from the system to finish querying SQL\*ReportWriter's tables.

**Action:** Make more memory available, if you can, or contact the system administrator for this machine.

**1555: There are duplicate panel numbers defined for text <in err msg>.**

**Cause:** All panels that belong to a particular text Objects and Type must have unique numbers. You may have mistyped the panel number.

**Action:** Use [Insert Record Above] or [Insert Record Below] and enter the panel number you wish. This will automatically renumber the existing panels.

**1558: ORACLE error occurred while trying to lock the SQL\*ReportWriter tables.**

**Cause:** You attempted to logon to ORACLE or lock the SQL\*ReportWriter tables under the same username/password as another user, while the other user held an exclusive lock on the SQL\*ReportWriter tables.

**Action:** Wait for ORACLE to release the other user's locks on the SQL\*ReportWriter tables before attempting to access them.

**1562: The frequency of field <in err msg> in text <in err msg> is inconsistent.**

**Cause:** You referenced the field in a text object where it is not allowed. These invalid object might include a Page Header or Page Footer, a group that falls in a different path of the query structure than the field's own path, a non-body group object in the same group as the field, or for non-computed fields, the body of a higher group.

**Action:** Make sure none of the above circumstances apply to your report.

**1563: The frequency of summary <in err msg> in text <in err msg> is inconsistent.**

**Cause:** You referenced the summary in a text object where it is not allowed. These invalid objects might include:

-A group that falls in a different path of the query structure other than the summary's own path.

-A Title or Trailer Page, Report Header or Footer, or Page Header or Footer whose Print Group is not Report.

-The group footer of the field being summarized, if the summary uses a running function.

-In a matrix report, a group footer belonging to the same group where the summary is referenced in a sub\_footer.

**Action:** Reference this field in a valid text object.

**1565: Text <in err msg> does not fit within the Width of the report.**

**Cause:** The boilerplate text and field/summary references in this text object add up to more columns than can fit between Margins of the report.

**Action:** You can do any of the following to correct this problem.

-Increase the Page Width or decrease the Left/Right Margins

-Decrease the Width of any fields or summaries that are referenced in this text object

-Edit the text object itself; reduce white space, use morelines, etc.

**1566: Text <in err msg> does not fit within the Height of the report.**

**Cause:** The boilerplate text and field/summary references in this text object adds up to more lines than can fit between the Top and Bottom Margins of the report.

**Action:** You can do any of the following to correct the problem

-Increase the Page Height or decrease the Top/Bottom Margins

-Delete some of the fields and summaries that are referenced in this text object

-Edit the text object itself; reduce white space, use fewer lines, etc.

**1567: Group <in err msg> on panel `<in err msg> does not fit within the Width of the report.**

**Cause:** The boilerplate text and field/summary references in this group text adds Up to more columns than can fit between the Left and Right Margins of the report.

**Action:** You can do any of the following to correct the problem:

-Increase the Page Width or decrease the Left/Right Margins.

-Decrease the Width of any fields or summaries that are referenced in this group text.

-Edit the group text reduce white space, use more lines, delete text, or remove field and summary references.

**1570: Invalid attempt to create a multi-panel report. The problem occurred while processing field <in err msg> on panel <in err msg>.**

**Cause:** You have specified a Relative Position of PANEL for a field whose group is not defined as multi-panelled.

**Action:** Either place an X in Multi-Panel for the field's group, or change the field's Relative Position to something other than PANEL.

**1571: Field <in err msg> on panel <in err msg> does not fit within the Height of the report**

**Cause:** The position of fields and text on this panel has taken up so much room that there is not enough room to fit the field on the page.

**Action:** Increase the Page Height on the Report Screen, or reposition the fields and text on the panel.

**1572 Field <in err msg> on panel <in err msg> does not fit within the Width of the report**

**Cause:** The position of fields and text on this panel has taken up so much room, there is not enough room to fit the field mentioned in the error message on the page.

**Action:** Increase the Page Width on the Report Screen, or reposition the fields and/or text on the panel.

**1573: This report exceeds the limit of 10 panels. The problem occurred while processing field <in err msg> on panel &err msg>.**

**Cause:** This report would require more than 10 panels to execute.

**Action:** There are many ways to reduce the number of panels, such as decreasing the width of fields and summaries, deleting panels or moving fields closer together, etc.

Consider the field identified in the error message when estimating how close you currently are to the 10 panel maximum. Estimate how close this field is to the end of the report.

**1578 Group <in err msg> on panel <in err msg> does not fit within the Height of the report.**

**Cause:** The size of each group text is alright, but taken together, they exceed the Page Height.

**Action:** Increase the Page Height on the Report Screen, or reduce the size of one or more texts belonging to this group.

**1579: Matrix group <in err msg> does not fit within the Height of the report.**

**Cause:** The size of each matrix group text is alright, but taken together, they exceed the Page Height.

**Action:** Increase the Page Height on the Report Screen, or reduce the size of one or more texts belonging to this matrix group.

**1580: Matrix group <in err msg> does not fit within the Width of the report.**

**Cause:** The size of each matrix group text is alright, but taken together, they exceed the Page Width.

**Action:** Increase the Page Width on the Report Screen, or reduce the size of one or more texts belonging to this matrix group.

**1582: A matrix report cannot have multiple panels.**

**Cause:** You defined more than one panel for this matrix report.

**Action:** Delete the extra panels.

**1583: Field <in err msg> cannot appear in both the 'Down' group and the 'Across' group in a matrix report.**

**Cause:** You referenced this field in one or more of the texts belonging to the 'Down' group as well as one or more of the texts belonging to the 'Across' group.

**Action:** Check the texts belonging to one of these groups on the Text Screen, and make sure that none of these texts reference this field.

**1584: The matrix does not fit between the margins of the report.**

**Cause:** The Page Header and Footer plus the Report Header and Footer exceed the Report Height or the Report Width, or the Top Margin plus the Bottom Margin equals the Report Height on the Report Screen.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of the matrix.

**1587: This report does not fit between the page margins.**

**Cause:** The page margins exceed the height and/or width of the report.

**Action::** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1588: The Report Header or Footer does not fit between the margins of the report**

**Cause** These text objects are too large, spanning beyond the report's margins.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1589: The Title Page does not fit between the margins of the report**

**Cause:** The Title Page is too large, spanning beyond the report's margins.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1590: The Trailer Page does not fit between the margins of the report**

**Cause:** The Trailer Page is too large, spanning beyond the report's margins.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1591:The Page Header does not fit between the margins of the report.**

**Cause:** The Page Header is too large, spanning beyond the report's margins.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text object s.

**1592: The Page Footer does not fit between the margins of the report-**

**Cause:** The Page Footer is too large, spanning beyond the report's margins.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1593: The Report Header does not fit between the margins of the report.**

**Cause:** The Report Header is too large, spanning beyond the report's margins.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1594: The Report Footer does not fit between the margins of the report.**

**Cause:** The Report Footer is too large, spanning beyond the report's margins

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1595: The Header of group <in err msg> does not fit between the margins of the report.**

**Cause:** This group header is too large, spanning beyond the report's margins.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1596: The Column Heading of group <in err msg> does not fit between the margins of the report**

**Cause:** This column heading is too large, spanning beyond the report's margins.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1597: The Body of group <in err msg> does not fit between the margins of the report.**

**Cause:** This group body is too large, spanning beyond the report's margins.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1598: The Footer of group <in err msg> does not fit between the margins of the report.**

**Cause:** This group footer is too large, spanning beyond the report's margins.

**Action:** Decrease the margins or increase the page size on the Text Screen, or decrease the size of other text objects.

**1601: Summary <in err msg> on panel <in err msg> extends beyond the left margin of group <in err msg>.**

**Cause:** The width of this summary is greater than the width between the right and left margins of the group. Since this summary is either center-or right-justified, SQL\*ReportWriter would encounter the left margin while printing it.

**Action:** Decrease the margins of that panel or increase the group's margin on the Text Screen, or decrease the summary's width on the Summary Screen.

**1602: Summary <in err msg> on panel <in err msg> extends beyond the right margin of group <in err msg>.**

**Cause:** The width of this summary is greater than the width between the right and left margins of the group. Since this summary is either center-or left-justified, SQL\*ReportWriter would encounter the right margin while printing it.

**Action:** Decrease the margins of that panel or increase the group's margin on the Text Screen, or decrease the summary's width on the Summary Screen.

**1604: Query <in err msg> cannot have more than one parent because this is not a matrix report.**

**Cause:** You defined a query structure where one query has two parent queries. However, none of the groups are matrix groups (i.e., have an X in Matrix Group on the Group Screen).

**Action:** Make Parent Query 2 blank for the query identified in the error message, or make this a matrix report.

**1605: Text <in err msg> cannot contain a &CR because it has no fields with variable Alignment.**

**Cause:** Hard returns (created by embedding&CR in the text) are only allowed in text objects that contain at least one field with an Align setting of Variable.

**Action:** Use a soft return (created by pressing the Return key) to achieve the effect of a hard return, or change the Align setting for one of the fields to Variable.

**1606: Query <in err msg> cannot reference Field <in err msg> because the field is not reset at a group whose query is a parent of <in err msg>.**

**Cause:** A field that is referenced in a query must have a Reset Group at, or above, the group that is created by the query.

**Action:** Press [List] and select a Reset Group (for the field you wish to reference) at or above the group created by the query.

**1607: Query <in err msg> cannot reference Summary <in err msg> because the summary is not reset at a group whose query is a parent of <in err msg>.**

**Cause:** A summary that is referenced in a query must have a Reset Group at, or above, the group that is created by the query.

**Action:** Press [List] and select a Reset Group (for the summary you wish to reference) that is at, or above the group that is created by the query.

**1608: Field <in err msg> cannot contain a compute function because its Source is a user exit.**

**Cause:** A Computed Value (Function) cannot be performed on a user exit.

**Action:** Either compute the function you wish in your user exit, or create a Summary on a field that has a Source of a user exit.

**1609: Query <in err msg> cannot reference a Field (<in err msg that belongs to a group in or after the group associated with the query.**

**Cause:** You cannot reference a Field in a query if the field belongs in, below, or to the right of the group created by the query.

**Action:** Add the field (the one that you want to reference) to a query that creates a group above, or to the left of the group that is created by the query (you want to add the reference to).

**1610: Query <in err msg> cannot reference a Summary (<in err msg>) whose Field belongs to a group in or after the group associated with the query.**

**Cause:** You cannot reference a Summary in a query if the field belongs in, below, or to the right of the group created by the query.

**Action:** Add the field (the one that you want to reference) to a query that creates a group above, or to the left of the group that is created by the query (you want to add the reference to).

**1611: Text <in err msg> cannot contain a variable field after a wrap field.**

**Cause:** Your text object contains one or more Variable fields below, or to the right of a Wrap field.

**Action:** Change the Align setting for the Wrap or Variable field(s) so that this rule is not violated.

**1612: Query <in err msg> cannot reference a Field (<in err msg>) that resets at PAGE.**

**Cause:** You cannot reference a Field in a query if the field has a Reset Group at PAGE.

**Action:** Press [List] and select a Reset Group that is not PAGE.

**1613: Query <in err msg> cannot reference a Summary (<in err msg>) that resets at PAGE.**

**Cause:** You cannot reference a Summary that has a Reset Group of PAGE in a query.

**Action:** Press [List] and select a Reset Group that is not PAGE.

**1614: Query <in err msg> cannot reference a Field (<in err msg>) with Source of &PAGE or &NUM\_PAGES.**

**Cause:** You tried to reference a Field in a query that has a Source of &KPAGE or &NUM\_PAGES.

**Action:** Press [List] and select a Source that is not &PAGE or &NUM\_PAGES.

**1615: Matrix query <in err msg> cannot reference a Field or Summary because it is not the cell group query.**

**Cause:** You can only reference Fields or Summaries in a query, if the query's default group has a Print Direction of Crosstab.

**Action:** Add the Field or Summary you wish to reference to the Query that has a default group with a Print Direction of Crosstab.

**1616: Query <in err msg> cannot have a parent query with no associated group.**

**Cause:** You created a parent-Child relationship between two queries (on the Query Screen) and then removed the group associated with the parent query (on the Group Screen).

**Action:** Create a group (on the Group Screen) for the parent query and associate it with the query.

**1617: Field <in err msg> cannot reset at Page and be referenced in variable length text object <in err msg>.**

**Cause:** You set the Reset Group for the computed field to PAGE, but the computed field is set to print in a text object that contains a field or summary with a Wrap or Variable alignment.

**Action:** Create a summary of the field and place it in the Page Header or Page Footer of the report.

**1618: Summary <in err msg> cannot reset at Page and be referenced in variable length text object <in err msg>.**

**Cause:** You set the Reset Group for the summary field to PAGE, but the summary is set to print in a text object that contains a field or summary field with a Wrap or Variable alignment.

**Action:** Create a summary of the summary field and place it in the Page Header or Page Footer of the report.

**1619: Source of field <field name> cannot contain a '#' without a user exit name.**

**Cause:** You did not include the name of the user exit after the '#' sign when you tried to create the user exit.

**Action:** Specify a valid user exit name after the '#' in the Source.

**1701: Field <in err msg> cannot be printed at a higher group.**

**Cause:** Afield cannot print in a higher group because it could possibly have multiple values, and SQL\*ReportWriter would not know which one to print.

Exceptions include calculated fields that would have a single discrete value if printed in the higher group.

**Action:** Delete the field references on the Text Screen from the group texts that are higher than the field's own group.

**1702: Field <in err msg> cannot reset at a lower level than its print level.**

**Cause:** The calculated field appears in a text object that is higher than the field's Reset Group. It does not make sense for the calculated field's accumulator to reset more often than the field actually prints.

**Actio:** Reference this calculated field in a text object that is at the same level or lower than the Reset Group.

**1703: field <in err msg> cannot reset at a lower level than its logical group.**

**Cause:** The calculated field's Reset Group is lower than the group that the field belongs to.

**Action:** Make the Reset Group the same level or higher than the field's group.

**1704: field <in err msg> cannot reset at a lower or equal level than its print level.**

**Cause:** The calculated field appears in a text object that is higher or at the same level as the field's Reset Group.

**Action:** Reference this calculated field in a text object that is lower than the Reset Group.

**1705: Field <in err msg> cannot reset at a lower or equal level than its logical group.**

**Cause:** The calculated field's Reset Group is lower or at the same level as the group that the field belongs to.

**Action:** Make the Reset Group higher than the field's group.

**1706:Summary <in err msg> cannot be printed at a higher group.**

**Cause:** The summary appears in a text object that is higher than the allowed.

**Action:** Reference this summary in a lower group.

**1707: Summary <in err msg> cannot reset at a lower level than its Reset Group.**

**Cause:** This summary appears in a text object that is higher or at the same level as the summary's Reset Group.

**Action:** Reference this summary in a lower group, or make its Reset Group higher.

**1708:Summary <in err msg> cannot reset at a lower level than its logical group.**

**Cause:** The summary's Reset Group is lower than the group that the field being summarized belongs to.

**Action:** Reference this summary in a lower group, or make its Reset Group higher.

**1709:Summary <in err msg> cannot reset at a lower or equal level than its print level.**

**Cause:** This summary appears in a text object that is higher than the summary's Reset Group.

**Action:** Reference this summary in a lower text object, or make its Reset Group higher.

**1710: Summary <in err msg> cannot reset at a lower or equal level than its logical group.**

**Cause:** The summary's Reset Group is lower or at the same level as the group that the field being summarized belongs to.

**Action:** Reference this summary in a lower group, or make its Reset Group higher.

**1712: The Source and the group for field <in err msg> cannot belong to queries in different paths.**

**Cause:** There is an inconsistency between the Source column for this field and the field's group.

The field's Source column must be from a query that is in the same path (of the query structure) as the field. The field's path is determined by the field's group. In turn, the group's path is determined by the group's query.

**Action:** Either change the Source column or the Group for the field on Field Screen 1. Make sure the Source and Group are from the same path of the query structure.

**1713: Field <in err msg> references an unknown source column.**

**Cause:** The Source for this field is currently undefined. This might result because a database link is missing, the underlying table definition has changed, or any other reason that prevents SQL\*ReportWriter from locating the Source originally reference&

**Action:** Either define this field in a valid way, or delete the field from your report definition.

**1714: Field <in err msg> references a Source whose query <in err msg>, has no groups.**

**Cause:** A field's Source must contain a group name that belongs to a query. You deleted a group that belonged to a query, but that group name is still in the Source column of the field (in the error message).

**Action:** Delete the field's Source, and select a valid Source from the List of values.

**1721: Cannot create a runfile.**

**Cause:** SQL\*ReportWriter tried to create or overwrite a rep file in a directory for which you do not have read/write permission.

**Action:** Change the permissions mode of the file or directory so that you can write rep files.

**1728: Break group <in err msg> must contain at least one database field.**

**Cause:** The group identified in the error message only includes fields that are based on the same columns as fields in higher groups, or only includes computed or summary fields. At least one field must be based on a column that has not been previously referenced.

**Action:** Delete the group in question, or add at least one valid field.

**1729: Field, summary, or parameter referenced by query <in err msg> no longer exists.**

**Cause:** You deleted a field, summary, or query and you did not remove it from the query.

**Action:** Delete the field, summary, or parameter from the query named in the error message. If you do not know which reference needs to be deleted, look at the fields, summaries, and parameters that are found on their perspective screens.

**1731: Missing closing quote in Source Column user exit for field <in err msg>.**

**Cause:** Your user exit is missing a quote. Each string that you begin with a quote must also end with a quote. You use the quotes when you do not want SQL\*ReportWriter to add that string's contents to the dependency list. If you wanted to pass an actual quote, use \".

**Action:** Add the missing quote, or delete the erroneous one.

**1732: Source &SQL statement for field <in err msg> contains a bind parameter that is not a Field, Summary, or Parameter.**

**Cause:** In your&SQL statement, you entered a bind parameter (any value preceded by a colon) that is not a Field, Summary, or Parameter.

**Action:** Make the value you entered a Field, Summary, or Parameter. If you did not intend to use a bind parameter, delete the colon located before the word in the&SQL statement.

**1733:Source Column user exit for field <field name> cannot reference a summary, <summary name>, with that field as its source field.**

**Cause:** A user exit references a summary whose source field is the user exit field itself.

**Action:** Remove the summary reference from the user exit, *or* change the summary's source field to another field.

**1751: Group <in err msg> references an unknown query.**

**Cause:** The query identified for this group is undefined.

**Action:** Move to the Group Screen and identify a valid query for the group, or move to the Query Screen and add or rename an appropriate query.

**1752: Field <in err msg> references an unknown query.**

**Cause:** The query associated with the Source column on the Field screen for this field is not defined.

**Action:** Move to the Field Screen and identify a valid Source, or move to the Query Screen and add, edit, or rename an appropriate query.

**1753: Field <in err msg> references an unknown group.**

**Cause:** The group associated with the field on the Field Screen is not defined.

**Action:** Move to the Field Screen and identify a valid group, or move to the Group Screen and add or rename an appropriate group.

**1754: Afield <in err msg> references an unknown reset group.**

**Cause:** The reset group associated with the computed field on the Field Screen is not defined.

**Action:** Move to the Field Screen, and identify a valid reset group, or move to the Group Screen and add or rename an appropriate group.

**1755: Summary <in err msg> references an unknown field.**

**Cause:** The field associated with the summary on the Summary Screen is not defined.

**Action:** Move to the Summary Screen and identify a valid field, or move to the Field Screen and add or rename an appropriate field.

**1756: Summary <in err msg> references an unknown print group.**

**Cause:** The print group associated with the summary on the Summary Screen is not defined.

**Action:** Move to the Summary Screen, and identify a valid print group, or move the Group Screen and add or rename an appropriate group.

**1757: Summary <in err msg> references an unknown reset group.**

**Cause:** The reset group associated with the summary on the Summary Screen is not defined.

**Action:** Move to the Summary Screen, and identify a valid reset group, or move the Group Screen and add or rename an appropriate group.

**1758: Query <in err msg> references an unknown parent query.**

**Cause:** The query associated with the parent query on the Query Screen is not defined.

**Action:** Identify a valid query for the parent query, or add or rename a query that matches the name in Parent Query.

**1760: Text <in err msg> references an unknown frequency group.**

**Cause:** The group identified in Frequency on the Text Screen for this text object is not defined on the Group Screen

**Action:** Move to Frequency on the Text Screen and identify a valid frequency group, or move the Group Screen and add or rename an appropriate group.

**1761: Foreign key <in err msg> references an unknown item**

**Cause:** The item referenced by the foreign key is not defined.

**Action:** Reference a valid item.

**1762: Foreign key <in err msg> references an unknown parent.**

**Cause:** The parent referenced by the foreign key is not defined.

**Action:** Reference a valid parent. To do so, press [List] while the cursor is in the Parent column of the Foreign key in question, and select a parent from the list.

**1763: Foreign key <in err msg> has an invalid local column position value.**

**Cause:** The local column position value referenced by the foreign key is not valid.

**Action:** Reference a valid value.

**1764: Foreign key <in err msg> has an invalid foreign column position value.**

**Cause:** The foreign column position value referenced by the foreign key is not valid.

**Action:** Reference a valid value.

**1771: Summary <in err msg> cannot reference a field with Data Type of PRT.**

**Cause:** You tried to create a summary of a field that has a Data Type of PRT.

**Action:** Press [List] and select a field from the list of valid values.

**1772: Summary <in err msg> cannot reference a field with Source of &DATE, &PAGE, or &NUM\_PAGES.**

**Cause:** You tried to create a Summary of a field that has a Source of &DATE, &PAGE, or &NUM\_PAGES.

**Action:** Press [List] and select a field from the list of valid values.

**1780: Invalid specification for a matrix report**

**Cause:** You used some - but not all - of the conventions for defining a matrix report. You must use all, or none of the conventions.

**Action:** If any group has an X in Matrix Group, then the report must have exactly 3 groups (each from a different query), and all 3 groups must have an X in Matrix Group. One group must have DOWN for Print Direction, another group must have ACROSS and the third group must have CROSSTAB. See Matrix Group for more information about creating a matrix report.

If you do not want this to be a matrix report, remove the X's from Matrix Group for all groups, and either change the Print Direction of the CROSSTAB group or delete it.

**1781: This report includes both matrix and non-matrix groups.**

**Cause:** Your report has at least one group where Matrix Group does not have an X, or at least one Print Direction of Down Across or Crosstab is specified, but all three Print Directions are specified.

**Action:** If any group has an X in Matrix Group, then the report must have exactly 3 groups (each from a different query), and all 3 groups must have an X in Matrix Group. One group must have DOWN for Print Direction, another group must have ACROSS and the third group must have CROSSTAB. See Matrix Group for more information about creating a matrix report.

**1783: The Print Direction for group <in err msg> and another group is inconsistent**

**Cause:** You have defined at least one DOWN (or DOWN/ACRmS) group and one ACROSS (or ACROSS/DOWN) group in this report.

**Action:** Make sure the Print Direction for all the groups in your report is DOWN (or DOWN/ACROSS) or all the groups are ACROSS (or ACROSS/DOWN).

If you are defining a matrix report, you should see the rules about defining matrix groups.

**1785: Wrap fields and Variable fields are not allowed in matrix, across, and across/down reports.**

**Cause:** One or more fields in the report has an Align setting (on Field Screen Three) of Wrap or Variable.

**Action:** Change the Align setting for the Wrap or Variable field(s) by pressing [List] and selecting one of the settings, or change the Print Direction for the group(s), by pressing [List] in the Print Direction entry field (on Group Screen One) and selecting Down or Down/Across for the setting.

**1786:Wrap fields and Variable fields are not allowed in multi-panel reports.**

**Cause:** You created a multi-panel report that contains a Wrap and/or Variable field.

**Action:** Choose another Align setting for the Wrap or Variable field(s) (press [List] to see other Align settings that are available), or modify the text panels so that there is only one panel for each text Object and Type.

**1787: Groups in a matrix report must have unique queries.**

**Cause:** You specified the same query for at least two groups in a matrix report. A matrix report must have three queries and three groups, and each group must assigned to a unique query.

**Action:** Change the Query assignment for the groups so that all three assignments are unique.

**14370: Your runfile was generated by a prior version of SQL\*ReportWriter.**

**Cause:** The runfile for this report has not been regenerated since a new version of SQL\*ReportWriter was installed at your site.

**Action:** Regenerate the report.

**14373: This runfile was generated on a different machine; its binary format is not compatible with the current environment.**

**Cause:** The runfile was generated on a computer that is not compatible with the one that you are currently using.

**Action:** You can do either of the following

1. Regenerate the runfile on this computer or a compatible one.
2. Move the existing runfile to the computer where it was originally generated (or a compatible computer), and run the report there.

**14377: You did not specify the name of a report**

**Cause:** You tried to run a report without specifying its name on the command line.

**Action:** Include the name of a valid report after the command. If the report name is not the first value on the command line, you must use the REPORT keyword (for example, REPORT=reportname).

**14378: You tried to open an invalid file.**

**Cause:** The file you referenced is incompatible with the format used by SQL\*ReportWriter.

**Action:** Regenerate the runfile.

**14381: You tried to execute an item that has not yet been defined.**

**Cause:** This condition should not normally occur.

**Action:** Contact Oracle Customer Support for assistance.

**14848: You used a date symbol in the format picture that was not recognized.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14849: You used a nonprintable character in the format picture.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14850: You left out the final quote in the format picture.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14851: You have used an invalid DAY in the format picture.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14856: You have left out punctuation from the input date that is supplied in the format picture.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14859: You have provided an input date shorter than the format picture string calls for.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14861: You have left out data that is called for by the format picture.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14863: You supplied a day of the month larger than the month includes.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14872: You did not provide a day-of-month number (1...31), as was expected in the input date.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14879: You did not provide a minute (0..59), as was expected in the input date's time.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14880: You did not provide a month number (1..12), as was expected in the input date.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14882: You did not provide a month name abbreviation (Ja.n...Dec), as was expected in the input date.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**14897: You did not provide the last two digits of the year number (0...99), as expected in the input date.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**15072: You tried to reference a field whose context is not active.**

**Cause:** This condition should not normally occur.

**Action:** Contact Oracle Customer Support for assistance.

**15107: Too many pages formatted. There is either too much data or an error in the report's definition.**

**Cause:** The report would have been more pages than allowed by the maximum value (this value is set in the PRODUCT\_PROFILE table).

**Action:** Modify your report definition to reduce the number of pages, or have your SQL\*ReportWriter administrator change the maximum for the PRODUCT\_PROFILE table.

**15126: This function does not do anything in this context.**

**Cause:** You tried to use a SQL\*ReportWriter function that only works in other situations.

**Action:** If you do not know which keys work, press [Key Help] for a list of the valid functions in this context.

**15127: There is no current record to delete.**

**Cause:** You pressed [Delete Record], but the cursor is not on a record.

**Action:** Place the cursor on the record you wish to delete, then press [Delete Record].

**15128: A list of values is not available here.**

**Cause:** There is no LIST for this attribute.

**Action:** If you need information about your options, acknowledge this error message, and press [Help].

**15131: You tried to enter the mode you are already in.**

**Cause:** you pressed [Query], even though you are already in Query Mode, or you pressed another mode key, even though you were already in that mode.

**Action:** Acknowledge the error message, and continue defining your report.

**15132: You tried to enter the mode you are already in.**

**Cause:** You pressed [Query], even though you are already in Query Mode, or you pressed another mode key, even though you were already in that mode.

**Action:** Acknowledge the error message, and continue defining your report.

**15133: You pressed [Fetch] before you entered Query Mode.**

**Cause:** [Fetch] only works in Query Mode.

**Action** If you want to search for specific text objects, press [Query] to enter Query Mode (a QUERY indicator appears on the status line while in Query Mode). Enter any criteria you want for the search. Then press [Fetch] to retrieve the texts from the SQL\*ReportWriter tables.

**15134: This function does not do anything when the Text Screen is in Query Mode.**

**Cause:** Many functions have no meaning in Query Mode.

**Action:** If you wish to use this function, leave Query Mode by pressing [Fetch]. The QUERY indicator on the status line disappears, and the screen is no longer in Query Mode.

**15140: The value you entered has the wrong data type.**

**Cause:** The data type that you entered is not valid in this context.

**Action:** Change the data type to a valid option.

**15146: The list of values does not currently have any items.**

**Cause:** You tried to enter a list of values, but the list of values associated with this field contains no records.

**Action:** Continue defining your report without accessing a list of values here.

**15148: This function does not do anything when the cursor is in a list of values.**

**Cause:** You used a SQL\*ReportWriter function that only works in other contexts.

**Action:** Press the spacebar or any alphanumeric key to acknowledge the message and use valid functions.

If you do not know which keys work, press [Key Help] for a list of the valid functions in this context.

**15151: This function does not do anything in this context.**

**Cause:** You used a SQL\*ReportWriter function that only works in other contexts.

**Action:** Press the spacebar or any alphanumeric key to acknowledge the message and use valid functions.

If you do not know which keys work, press [Key Help] for a list of the valid functions in this context.

**15164: NO records found.**

**Cause:** You probably entered criteria in Query Mode that did not correspond to any text objects in the database.

**Action:** Modify the criteria and press [Fetch] again.

**15166: An attempt was made to set the value of a form/report field in a group that does not allow updates.**

**Cause:** You tried to set the value of a field that has a Source of a column from a SELECT statement.

**Action:** Create a new field with the same Source below the field you want to change, and set the value of the new field.

**15168: You tried to undelete a record when there was no deleted record.**

**Cause:** You pressed [Undelete Record] when there was no record in the buffer.

**Action:** Enter the value(s) you want directly.

**15170: You cannot use this print function from within a list of values.**

**Cause:** You pressed either [Print Page] or [Print Report] while the cursor was in a list of values.

**Action:** You can press [Print Screen] to capture the current screen into a default log file. Otherwise, select an item from the current list of values or press [Undo].

**19682: You pressed a key that does not select an item in this menu.**

**Cause:** The cursor is in a menu with a list of items. You pressed a key that does not select any of these items.

**Action:** There are two ways to choose an item in the menu:

1. Use the arrow keys to move the highlight bar to the item you want, then press [Select].
2. Press an alphabetic key that corresponds to the first letter of the item's name.

**19776: You have entered too many digits to the right of the decimal point.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**19777: The value you entered is not numeric.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**19791: You have typed an invalid number picture symbol.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**19792: You have two floating characters in one number picture.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**19793: You have the float character after a zero-suppressible digit.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**19794: YOU have a zero-suppressible digit after a regular digit.**

**Cause:** You entered a format that SQL\*ReportWriter does not recognize.

**Action:** Enter a valid Display Format, or leave this attribute blank.

**19840: File open error null or illegal filename.**

**Cause:** This is an abnormal condition.

**Action:** Contact Oracle Customer Support for assistance.

**19841: File open error: directory is protected from file access.**

**Cause:** SQL\*ReportWriter tried to open a file with insufficient directory privileges.

**Action:** Upgrade the directory privileges.

**19842: File open error: file is read-protected.**

**Cause:** SQL\*ReportWriter tried to open a file that is read-protected.

**Action:** Upgrade the file privileges for SQL\*ReportWriter.

**19843: File open error: file is write-protected.**

**Cause:** SQL\*ReportWriter tried to open a file that is write-protected.

**Action:** Upgrade the file privileges for SQL\*ReportWriter.

**19844: File open error file was not found.**

**Cause:** SQL\*ReportWriter tried to open a file, but could not find it in the system.

**Action:** Identify the location of the file for SQL\*ReportWriter.

**19846: File open error unable to open file.**

**Cause:** SQL\*ReportWriter tried to open a file, but was unsuccessful. The file may be defective or there could be a system problem.

**Action:** Make sure your system is operating correctly. If it is, determine which file SQL\*ReportWriter tried to open and make sure appropriate privileges are available, or investigate other causes.

**19848: Unable to create the file because it was already opened.**

**Cause:** The name of the output file and the name of the runfile are the same.

**Action:** Change the name of the output file on the Report Screen.

**19849: Error attempting to share lock the file.**

**Cause:** SQL\*ReportWriter could not lock a file using a share lock because the file had already been locked using an exclusive lock. This error might occur if one user has opened a report and another user attempts to run the same report using RUNREP.

**Action:** Wait until the other user has finished modifying the file and the exclusive lock has been released before attempting to access the same file.

**19850: A network error has occurred.**

**Cause:** SQL\*ReportWriter encountered a network error while processing a report definition.

**Action:** Identify the source of the network problem and have it fixed.

**19851: An invalid file pathname was specified.**

**Cause:** SQL\*ReportWriter specified a pathname that is not valid on this system.

**Action:** Identify the valid pathname.

**19852: The maximum number of files that can be open at one time has been exceeded.**

**Cause:** SQL\*ReportWriter is unable to open anymore files to process this report definition.

**Action:** The safest way to close files is to exit from SQL\*ReportWriter, then log back on and continue defining your report.

**19853: Unable to position to user-specified record.**

**Cause:** SQL\*ReportWriter tried to position to a byte in a file and the operating system refused. 'l' is most likely a disk or operating system error.

**Action:** If this error occurs repeatedly on one particular report, it is an internal error. Contact Oracle Customer Support.

**19854: Unable to write to file. File system could be full.**

**Cause:** The file system might be out of room, or you may not have adequate privileges to write to a file.

**Action:** Free up space or create sufficient privileges for SQL\*ReportWriter to write files.

**19872: The printer file specified was not a supported version.**

**Cause:** You referenced a printer definition file from a previous version of SQL\*ReportWriter.

**Action:** On the command line, recompile the printer description file using the PRINTDEF command. (Note You will need to recompile all printer description files that you created with the older version(s) of SQL\*ReportWriter. You will not be required to do this, however, when using future versions of SQL\*ReportWriter-that is, from Version 1.1.10 on.

**19875: The printer file specified was not a valid printer description (printdef) file.**

**Cause:** You referenced a printdef file that SQL\*ReportWriter cannot use.

**Action:** Identify the valid printer description files for your system, and reference one of them. If you are unsure about the location of these files, check the SQL\*ReportWriter installation instructions for your system.

**20706: Where is no more room to enter any more characters.**

**Cause:** You tried to enter more characters when this attribute already had the maximum number allowed.

**Action:** Enter a character string that fits within the width of the attribute.

**20707: You cannot write text to a file until you copy the text to the buffer.**

**Cause:** You pressed [Write File], but there was nothing in the text buffer.

**Action:** To insert a character string into the text buffer, press [Mark], move the cursor to the end of the string, then press [Cut] or [Copy]. You can then press [Write File], and SQL\*ReportWriter will prompt you for the name of the file.

**20709: You typed an invalid repeat count or octal number.**

**Cause:** The value entered for [Repeat] must be an integer between 1 and 99999.

**Action:** Enter a number in this range and press [Select]; otherwise, press [Undo].

**20710: You cannot edit text in this context.**

**Cause:** You tried using one of the editing functions, but there is nothing to edit in the current context.

**Action:** If you do not know which keys work in this context, press [Key Help] for a list of the valid functions.

**20711: You pressed an invalid key or sequence of keys.**

**Cause:** You pressed a key or a combination of keys that has no meaning for the terminal file that you are currently using. You may have done this unintentionally.

**Action:** Press the spacebar or any alphanumeric key to acknowledge the message and use valid functions.

If you do not know which keys work, press [Key Help] for a list of the valid functions in this context.

**20717: This function does not do anything in this context.**

**Cause:** You tried using a SQL\*ReportWriter function that is not relevant in the current situation.

**Action:** If you do not know which keys work, press [Key Help] for a list of the valid functions in this context.

**20719: You pressed [Undelete word], but there is no word to undelete.**

**Cause:** [Undelete Word] does not do anything when the word buffer is empty.

**Action:** Continue defining your report.

**20720: You pressed [Undelete Line], but there is no line to undelete.**

**Cause:** [Undelete Line] does not do anything when the line buffer is empty.

**Action:** Continue defining your report.

**20721: You pressed [Paste], but the paste buffer is currently empty.**

**Cause:** [Paste] does not do anything when the paste buffer is empty.

**Action:** First [Copy] or [Cut] a character string into the paste buffer, then move the cursor to the target location and press [Paste].

**20769: Task toggling. Press [Help] for more information.**

**Cause:** SQL\*ReportWriter has stacked multiple tasks. This is an abnormal condition.

**Action:** Contact Oracle Customer Support for assistance.

**20779: You have specified an invalid entry point. Press [Help] for more information.**

**Cause:** This is an abnormal condition.

**Action:** Contact Oracle Customer Support for assistance.

**20843: The file you specified is not a valid terminal definition file.**

**Cause:** You referenced a terminal definition file that does not exist.

**Action:** Make sure the name and path you specified for the terminal definition file are correct. If you are unsure what the default path is for the "term" parameter, check the Installation and User's Guide for your operating environment.

**20844: You cannot use this function until you leave the current mode.**

**Cause:** The SQL\*ReportWriter function that you tried to use only works in other contexts.

**Action:** If you do not know which keys work, press [Key Help] for a list of the valid functions in this context.

**20849: This function does not do anything in this context.**

**Cause:** The SQL\*ReportWriter function that you tried to use only works in other contexts.

**Action:** If you do not know which keys work, press [Key Help] for a list of the valid functions in this context.

**20853: Screen I/O in batch mode is not allowed.**

**Cause:** You executed the RUNREP command with BATCH=YES and DESTYPE=Screen, or BATCH=YES while the Run-time Parameter Form displayed a DESTYPE of Screen. When BATCH=YES, the report cannot be sent to the screen (DESTYPE cannot be Screen).

**Action:** Remove DESTYPE=Screen from the command line, or use SQLREP to modify the Parameter Screen so that DESTYPE is Printer, File, Sysout, or Mail.

**20896: No more name was specified for the bind variable of the WHERE clause.**

**Cause:** This is an abnormal condition.

**Action:** Contact Oracle Customer Support for assistance.

**20944: Cannot display the proper error message. Press [Help] for further information.**

**Cause:** SQL\*ReportWriter is unable to present an error message.

**Action** Contact Oracle Customer Support for assistance.

**20945: Error in command file.**

**Cause:** In the command file, a value (in Keyword= value) was invalid. This could be because a) a value did not begin with a letter, or b) a value contained characters other than alphanumeric characters (and underscores).

**Action:** Look in the command file, find the error, and modify it so that it no longer contains an invalid value as described above.

**20946A command line parameter was neither YES nor NO.**

**Cause:** On the command line you entered READONLY=value, BATCH=value, or PARAMFORM=value, where value was not YES or NO.

**Action:** Use YES or NO for the value of these keywords (for example, BATCH=YES).

**20960: You cannot enter Window Mode in this context.**

**Cause:** You pressed [Window] in a context where there is no window.

**Action** Do not use this function in this context. Only press [Window] if you are looking at output and want to browse downward or rightward on the page.

**20961: This function does not do anything in Window Mode.**

**Cause:** You used a SQL\*ReportWriter function that has no meaning in Window Mode.

**Action** If you do not know which keys work, press [Key Help] for a list of the valid functions in this context.

**21378: Printer not in printdef database.**

**Cause:** You entered the PRINTDEF command, specifying the name of a printer whose driver is not defined in the printer definition file.

**Action:** Specify another printer whose driver is defined in the printer definition file, or create a new printer definition for the printer in printdef.dat (or pstscrpt.dat) and run PRINTDEF to compile a new printer driver.

**2163: Invalid argument for keyword DESTYPE. Valid options are SCREEN, FILE, PRINTER, and MAIL.**

**Cause:** You provided an argument for the parameter DESTYPE that was not valid.

**Action:** Specify DESTYPE=SCREEN (or FILE, PRINTER, or MAIL); otherwise, do not use this parameter.

**21636: Unable to open the printer description file. Check the DESFORMAT parameter.**

**Cause:** The printer description file you referenced cannot be found in the relevant path, or it exists but SQL\*ReportWriter is unable to open it.

**Action:** Make sure you have identified a valid filename and path. If this information is correct, check restrictions that would prevent SQL\*ReportWriter from opening this file in your operating environment.

**28679: This SELECT Statement contains a word that is too long.**

**Cause:** You entered a word in your SELECT Statement that is too large to be processed by SQL\*ReportWriter.

**Action:** Shorten the name.

**28681: This SELECT Statement contains an invalid character.**

**Cause:** You entered a character in your SELECT Statement that SQL\*ReportWriter does not allow.

**Action:** Identify this invalid character, and remove it from your SELECT Statement.

**29010: You had a system error on input to or output from a file.**

**Cause:** This error could have been caused by one of three things: you are running out of disk space; your disk is malfunctioning; there is an internal error in SQL\*ReportWriter.

**Action:** Assign more disk space, or repair the disk if appropriate. Otherwise, contact Oracle Customer Support for assistance.

**29013:Not enough memory to finish this task.**

**Cause:** SQL\*ReportWriter cannot obtain enough memory from the system to complete the current task.

**Action:** Make more memory available, if you can, or ask your system administrator for assistance.

**29014: Requested a block of memory larger than 64K bytes. Press [Help] for more information.**

**Cause:** This condition should not normally occur.

**Action:** Contact Oracle Customer Support for assistance.

**29015:There was an error attempting to go to the operating system command line processor. Press [Help] for more information.**

**Cause:** This condition should, not normally occur.

**Action:** Contact Oracle Customer Support for assistance.

**29016: Cannot perform requested print operation.**

**Cause:** SQL\*ReportWriter cannot complete the current print task.

**Action:** Make sure your environment is properly configured for printing and try again.

**29017: Unable to bring up the error message file.**

**Cause:** SQL\*ReportWriter encountered an error while attempting to create a temporary virtual memory file.

**Action:** Check the permission mode on the default temporary directory. Make sure you have permission to write temporary virtual memory files in the directory.

**29052: Report size is not large enough. Try making it larger or taking “Repeat on Overflow” off of the same text items.**

**Cause:** The size of the report is not large enough. This causes overflow to occur but overflowing to the next page still does not allow the text objects to fit completely onto the Page.

**Action:** Enlarge the report or remove “Repeat on overflow” for some of the text objects. Start with the Body text objects first.

## **Abnormal Condition Errors**

Because SQL\*ReportWriter performs a large number of internal checks (more than sixty), many different types of abnormal conditions may occur. Rather than document each of these abnormal conditions, this section is provided to tell you what to do in case you receive an abnormal condition error.

**Abnormal Condition: One of two problems caused this abnormal condition:**

1. The likely cause: someone has by-passed the user interface, and has changed the data in SQL\*ReportWriter’s tables directly. An invalid value in these tables has prevented SQL\*ReportWriter from completing the task that you requested.

Alternatively, you may have detected a bug in SQL\*ReportWriter. If so, please write down the text you see in the error message, the circumstances that led to the error message, and forward this information to Oracle by calling or sending a letter to Customer Support.

**Abnormal Condition: SQL\*ReportWriterBug: The error message you received only occurs when SQL\*ReportWriter detects a bug in itself. Please do the following:**

1. Write down the text you see in the error message.
2. Write down the circumstances that led to the error message. Recall the actions you took, which report objects were involved (if any), and the exact sequence of keys that you pressed.
3. Forward this information to Oracle by calling or sending a letter to Customer Support.

### **Referenced Object Unknown**

**Cause:** An object references another object that no longer exists, probably because you did not select the "Cascade" option when deleting something.

**Action:** Update the report definition so that no objects reference a deleted object.

## GLOSSARY

**action menu** The Action Menu lists a set of operations to perform on the selected report. The actions are: New, Open, Copy, Rename, Drop, Generate, Execute, and Quit.

**across/down** A print direction that first prints each record of the group to the right of the previous record until the right margin of the page is reached. Then, records will printed below the left-most record on the same page.

**alert box** A pop-up window which asks for confirmation of an action.

**alias** A temporary name used in a SQL statement to reference a table or a column expression.

**align** (Field Screens) The values of fields may be aligned to the left, right, or center of the defined width of a field. Fields can also be wrapped, on word boundaries, onto multiple lines; or be variable width (i.e., have trailing spaces truncated).

**ancestor group** A group that is above another group, either directly or indirectly.

**ARRAYSIZE** A command line argument which allows you to specify the size of the array in kilobytes for use with the ORACLE array interface.

**argument** An expression within the parentheses of a function, supplying a value for the function to operate on. For example, on the command line, `KEYWORD=VALUE` is an argument.

**attribute** See setting.

**body** The text that is repeated for every record of a group.

**bottom margin** (Report Screen) The number of lines that SQL\*ReptWriter skips from the bottom of every page before outputting any text or data, including footers.

**break report** A report that consists of more than one group, where the groups are placed side by side. This is also known as an outline report.

**browse mode** Allows you to view report output on the screen page by page.

**call interface** An interface that will allow RUNREP and/or SQLREP to be linked with other ORACLE products, such as SQL\*Menu, or a user-written program.

**child query** When defining a master/detail report, by defining multiple tables, the child query retrieves detail information for values retrieved by the parent query.

**CMDFILE** A command line argument which allows you to specify a file which contains a set of arguments for SQLREP or RUNREP.

**column expression** An expression in a SELECT statement that defines the data that is retrieved from the database (see Field). May be a column name or valid SQL expression referencing a column name.

**column header** A type of text. By default, a column header contains the labels for the fields in the associated group.

**command line** Operating system command line. SQL\*ReportWriter can be invoked from the operating system command line using a number of parameters. See Chapters 3 and 4.

**comment** (Report screen) The comment setting provides a space to enter descriptive information about a report.

**compile** See generate.

**computed field** (Field Screens) Used to perform report-time calculations on data retrieved from the database. These calculations augment the kinds of calculations that can be done directly with a SELECT statement.

**COPIES** A command line argument which allows you to specify the number of copies of the report to print.

**crosstab** A print direction that prints each record of a group as the cell in a matrix. See Matrix Report.

**cursor** The lighted rectangle or bar on the terminal's screen which shows where the next character entered will appear.

**datatype** (Field, Summary Screens) Specifies the type of data contained in a field. May be CHAR, NUMBER, DATE, or PRT.

**date** One of ORACLE's data types. A date column may contain a date and time between January 1,4712 BC to December 31,4712 AD.

**default** A value that is used if no alternative is specified.

**DESFORMAT** A command line argument which allows you to specify the file that describes the characteristics of the printer named in DESNAME.

**DESNAME** A command line argument which allows you to specify the file or printer to which the report output will be written.

**DESTYPE** A command line argument which allows you to specify the type of output device that will receive report output.

**dialog box** A window asking the user to enter some piece of information necessary to complete an operation. Dialog boxes are used extensively in the Action Menu.

**display format** (Field, Summary, and Parameter Screens) Defines the appearance of the value of a field.

**down/across** A print direction that first prints each record of the group below the previous record until the bottom of the page is reached. Then, records will be printed to the right of the top-most record on the same page.

**DUMPREP** Command which unloads a report definition into an ASCII export file. This file can then be moved to another computer. This command is used in conjunction with the LOADREP command.

**execute** (Action Menu) Produces the report output.

**field** Container for the data retrieved by a particular column expression in a query. Define settings to control how the value in a field is displayed.

**field label** (Field Screens) Text that appears by default on the report output to describe a column of fields in a report.

**field name** (Field Screens) The name of the field. You must name every field for reference in other parts of the report definition. (See SQL Naming Rules.)

**field spacing** (Group Screens) Specifies the default number of blank spaces between all fields in the specified group for Down and Down/Across groups. Specifies the default number of blank lines between all fields in the specified group for Across and Across/Down groups.

**field width** (Field Screens) Specifies the width for each field.

**fields across** (Group Screens) Specifies the maximum number of fields in a group that will be placed on a single line of a panel by default for Down and Down/Across groups. Specifies the maximum number of fields that will be placed in a single column for Across and Across/Down groups.

**footer** A text type that prints after a group (group footer), at the bottom of each page (page footer), or at the end of the Report (report footer).

**frequency** (Text Screen) Defines how often the column heading will be printed.

**FROM** Required clause of the SELECT' statement that identifies the tables from which data is selected.

**function** (Field and Summary Screens) Computes the value of a field or summary.

**generate** (Action Menu) Creates a runfile enabling reports to be run via RUNREP.

**GENREP** A command which generates a runfile even in the absence of a database. This command is used in conjunction with RUNREP, which runs a report.

**group** Groups define sections and/or subsections of reports, and are also used for subtotals and totals.

**GROUP BY** Optional clause of a SELECT statement which identifies the level of summarization to be performed before fetching data from the database.

**group list** The list of groups found on the Group Settings screens.

**group name** (Group Screens) The name of a group. Groups are named so they can be referenced by other objects.

**group's query** Refers to the query associated with the group on the group settings screen.

**header** A text type that precedes the group (group header, prints at the top of a page (page header), or at the beginning of the report (report header).

**highlight** (Field, Group, and Text Screens) On the Field Screens, individual fields can be highlighted in the label area. Also on this screen, conditional highlighting can occur using user exits On the Group Screen, fields in a group can be highlighted. On the Text Screen, Mark and Highlight enable the highlighting of anything parts of fields or text, to entire reports. See Printer Control codes.

**index** An alphabetical listing of the topics available in the HELP system.

**join** Combining data from two (or more) tables in a single SELECT statement.

**justification** (Text Screen) The alignment of text, either, right, left, or center justified.

**label** (highlight) (Field Screens) Specifies the highlighting style for the label of a field.

**label position** (Group Screens) The position in which to print the field labels in the Body or Column Heading.

**left margin** (Report Screen) The number of spaces that SQL\*ReportWriter skips from the left of every page before outputting data.

**lines before** (Group, Field, Text Screens) The number of vertical lines to insert before printing an object.

**list of values** A list of the valid values in the current context. In some cases, the LIST shows existing names which cannot be used.

**LOADREP** Executable command which loads an ASCII export file into an ORACLE database. This is used in conjunction with the DUMPREP command.

**LOGFILE** A command line argument which allows you to identify the file used by the interactive print commands [Print Page], [Print Screen], and [Print Report].

**margin** See Left Margin and Right Margin.

**matching columns** (Query Screen) Information needed to link the data in two or more queries together. See Parent Query.

**matrix** (Group Screens) Indicates that a group is part of a matrix report.

**matrix report** Across-tabulation of data from three groups. The values in one group form the rows, the values from the second group form the columns, and the values from the third group form the cells.

**message line** The line at the bottom of the screen where SQL\*ReportWriter displays hints and other messages, including error messages

**multi-panel** (Group Screens) Specifies whether all fields in a group should remain on the same panel.

**object** An item in a report, for example, groups and fields, pages, and the report as a whole.

**ORDER BY** Clause used in a SELECT statement to specify the order in which the results of a query are to be displayed.

**output** Results of a report definition after it is executed. Output can be displayed on a screen, stored in a file, or printed on a hard copy.

**overflow** Overflow occurs when all of the records of a group do not fit on a single page..

**page** The area on which the report objects are placed.

**page break** (Group Screens) Controls whether or not to generate a page break.

**page height** (Report Screen) Specifies the height of the physical page in lines.

**page width** (Report Screen) The width of the physical page in character spaces.

**Panel** A panel is a chunk of text. Each text object may consist of one or more panels, each of which appear on a separate page.

**panel number** (Text Screen) The number of the panel on which to print text.

**PARAMFORM** A command line argument which allows you to display or suppress the Run-time Parameter Form when you execute a report.

**parent query** (Query Screen) When defining a master/detail report, by joining multiple tables, the parent query retrieves master information for values retrieved by the child query.

**PRINTDEF** Executable command which is used to create a new printer definition.

**print direction** (Group Screens) The direction in which each record of the group prints relative to the prior record.

**print group** (Summary and Field Screens) The Print Group determines how often the summary prints.

**printer control codes** Codes which enable users to use additional capabilities of printers.

**product\_profile** A table which enables printouts to be limited to a certain page limit for specified users.

**query** A SQL SELECT statement that retrieves information from one or more tables or views.

**query name** (Query and Group Screens) Name that identifies queries to be referenced on Group Settings or as the parent of another query.

**quit** (Action Menu) Option from the List of Actions that terminates a session and returns the user to the operating system.

**read consistency** The ability to query the data throughout a database as of a single point in time. This is needed to ensure that all the data in a report are internally consistent.

**READONLY** A command line argument which allows you to request read consistency without locking for a multiple query report by locking the tables in SHARE MODE.

**record** A record of a group represents a row from a query. Adding a group manually creates a mock record for each distinct set of values of the fields in a group.

**record spacing** (Group Screens) The number of blank lines or spaces between the records in a group.

**relative position** (Group, Field and Text Screens) The position of a group or field relative to the previous group or field. With regard to text, the location of text relative to the previous object.

**rename** (Action Menu) Used to change the name of a report.

**repeat** (Field Screens) This causes the field to appear on all panels of the report.

**repeat on page overflow** (Text Screen) Specifies whether a text object repeats on subsequent pages if the data overflows.

**REPORT** A command line argument which allows you to specify the report to execute.

**reset group** (Field and Summary Screens) The group at which the value of a function is to be reset too.

**right margin** (Report Screens) The number of spaces that SQL\*ReportWriter skips from the right of every page before outputting data. Note that if the group has a prior sibling of its own, the right boundary of that sibling will be used for the margin of the texts.

**row** One set of fields in a table.

**run** See execute.

**runfile** A compact representation of the report definition which is used at runtime to control the output of the report.

**running summary** Every summary function has a running version which returns cumulative values of the function between reset points.

**RUNREP** Operating system command that executes previously defined reports.

**setting** An area in which you specify a value for a SQL\*ReportWriter object characteristic.

**SELECT statement** (Query Screen) A SQL statement used to select rows and columns from one or more database tables.

**sibling** A group that is below another group, either directly or indirectly.

**skip** (Field Screens, Parameter Screens) Specifies whether SQL\*ReportWriter will output or suppress the indicated field in the location implied by the field settings. If enabled, the field will appear only when manually added to text objects. Also used to suppress parameters from appearing on the Run-time Parameter Form.

**source column** (Field Screens) A column name, expression, or alias appearing in a SELECT statement.

**source query** The query that produced the source column for the current field.

**spaces before** (Group, Field and Text Screens) The number of horizontal column positions to insert before printing an object.

**SQL** Standard interface for storing and retrieving information in a relational database. SQL is an acronym for "Structured Query Language."

**SQLREP** Operating system command that invokes SQL\*ReportWriter and allows you to define and execute reports.

**status** (Text Screen) Non-enterable field which indicates whether the current text is "Default" or "Edited."

**subtotal** See summary.

**summary** Summaries, or summary fields are used to compute subtotals, grand totals, running totals, and other summarizations of the data in a report.

**summary field** Field containing data derived by SQL\*ReportWriter using one of the summary functions. Summary fields are displayed by default in the group footer.

**summary name** (Summary Screens) The name given to the field being summarized for reference in other parts of the report definition.

**system variable** Variables that are provided by SQL\*ReportWriter (i.e., &DATE, &NUM\_PAGES, &PAGE\_NUM).

**system-owned tables** When individual users do not own their own SQL\*ReportWriter tables: the users share their tables with other users.

**TERM** A command line argument which allows you to specify a file that describes the terminal from which you are using SQLREP or RUNREP.

**text** (Text Screen) Texts contain field references and literal values which appear at various points in your report. See Text Type.

**Text Type** (Text Screen) Each group, and the report as a whole, has several text objects associated with it. Each text object has a type that defines where it will appear in the report output. Valid group types are header, footer, body, column heading and subfoot. Valid report text objects are title page, trailer page, report header and footer, and page header and footer.

**title page** A piece of text that appears on a separate page from other texts at the beginning of a report.

**toggle** Switch that can be turned on or off to control the behavior of report output or the definition process. For example, you can toggle between insert and overwrite mode when entering data.

**top margin** (Report Screen) The number of lines that SQL\*ReportWriter skips from the top of every page before outputting any text or data, including headings.

**trailer page** Text object which appears at the end of a report.

**USERID** A command line argument which allows you to specify your ORACLE username and password with an optional SQL\*Net address.

**user exit** Away in which to pass arguments from SQL\*ReportWriter to another computer program and, if desired, pass new or modified arguments to SQL\*ReportWnter.

**user-owned tables** When individual users own their own SQL\*ReportWriter tables the users do not share their tables with other users.

**view** A virtual table whose rows do not actually exist in the database. A virtual table is based on a table that is physically stored in the database.

**window mode** Allows you to view report output on the screen a line at a time. This is used for output that is less than one page but greater than one screen.

**width** (Field, and Text Screens) The width of a field, page, or panel. Width is measured in characters a space is a character.

**word wrap** The fittingg, or wrapping, of a field'scontents within a defined width (the width is fixed; the depth is dependent upon the number of characters the field contains)

**zoom** Allows you to expand the space of the current field so that you can edit it more easily. To return the field to its normal size, press any key that would normally take you out of that field.

## INDEX

### !

- %Total function
  - for computed fields 6-20
  - for summaries 7-5
- &CR (hard returns) 6-16, 8-13 to 8-14
- &DATE 8-12, 10-8
- &NUM\_PAGES 8-12
- &PAGE 8-12
- &SQL 6-4 to 6-6

### A

- Abnormal condition errors I-65
- Above 5-11
- Accept A-6
- Access privileges
  - granting 9-4
  - revoking 9-4
- Across 5-5
- Across/Down 5-5
- Action Menu
  - copy 3-5
  - description of 3-1
  - drop 3-7
  - execute 3-8
  - generate 3-9
  - new 3-3
  - open 3-4
  - rename 3-6

### Add

- newline A-2
- tab A-2
- Administration B-1
  - controlling access to other users' reports B-5
  - controlling access to SQL\*ReportWriter B-4
  - controlling access via
    - PRODUCT\_PROFILE B-5
  - creating printer drivers H-2
  - dropping reports B-1
  - establishing security B-1, B-4
  - files open during run-time B-7
  - installing a userid B-1, B-3
  - limiting the number of output pages B-5
  - memory requirements B-1
  - storage requirements B-1
  - upgrading from previous versions B-2
- Alert box 1-11
  - when deleting fields 6-2
  - when deleting groups 5-2
  - when deleting queries 4-2
- Align setting 6-15
- Always
  - page break 5-6
- ARRAYSIZE
  - for RUNREP 2-4
  - for SQLREP 1-3
- Avg function
  - for computed fields 6-20
  - for summaries 7-5

## B

### BATCH

for RUNREP 2-7

Beginning of line A-2

Below 6-13

relative position for text 8-9

relative position of fields 6-13

relative position of groups 5-7

Bind parameter 4-7

creating 10-2

definition of 10-2

examples in SQL clauses 10-3

Body 8-2

Bottom Margin setting 9-3

Browse mode 2-9

### BUFFERS

for RUNREP 2-6

for SQLREP 1-5

## C

Call interface D-1

Cancel application (Undo) A-7

Cascade

fields 6-2

groups 5-2

queries 4-3

cell 5-6

Center

field alignment 6-15

text alignment 8-11

Child Columns setting 4-9

CMDFILE

for RUNREP 2-4

for SQLREP 1-3

Column header

definition of 8-2

Command line interface

RUNREP 2-3

SQLREP 1-2

Comments setting 9-4

Computed Value setting

function 6-20

reset group 6-21

Conditional page break 5-6

Conditionally highlighting fields E-12

Converting

from V1.0 to V1.1 chart 2-2

from V1.0 to V1.1 using MOVEREP 2-13

reports from V1.0 to V1.1 B-2

COPIES

for RUNREP 2-5

for SQLREP 1-4

number of reports to print 10-9

system parameter 10-8

Copying reports

access privileges 9-5

functionality-see Access privileges 9-5

using the action menu 3-5

Copying text 1-15

Count function

for computed fields 6-20

for summaries 7-5

Creating

new reports 3-3

security B-4

Creating query objects 4-2

CREATION, column in the IAPXTB

table E-10

Crosstab 5-5 to 5-6

Cumulative functions

See Running functions

CURRENCY

for RUNREP 2-5

for SQLREP 1-4

system parameter 10-8

Currency setting 10-10

Current Date 10-8

Cut A-3

## D

- Data Type setting
  - for fields 6-10
  - for parameters 10-7
  - for summaries 7-6
- Database storage requirements B-6
- DATE
  - datatype 6-10
  - formats 6-12
  - system variable 8-12
- DDL
  - see SQL
- DECIMAL
  - for RUNREP 2-6
  - for SQLREP 1-5
  - system Parameter 10-8
- Decimal setting 10-10
- Default
  - logon 1-3
  - printer drivers H-2
  - query objects 4-3
- Default Value setting 10-8
  - current date 10-8
- DELETE
  - backward A-2
  - character A-2
  - line A-2
  - record A-4
  - to beginning of line A-2
  - to end of line A-2
  - word A-2
  - See also &SQL
- Delete buffer
  - See BUFFERS
- Deleting
  - query objects 4-2
- Deleting reports
  - see Drop
- DESFORMAT
  - default printer drivers H-2
  - for RUNREP 2-5
  - for SQLREP 1-4
  - system parameter 10-8

- DESNAME
  - for RUNREP 2-5
  - for SQLREP 1-4
  - system parameter 10-8
- DESTYPE
  - for RUNREP 2-5
  - for SQLREP 1-3
  - system parameter 10-8
- dfltrep.log 1-5, 2-6
- Dialog box 1-11
- Disk space requirements B-5
- Display Format setting
  - setting description for fields 6-11
  - setting description for summaries
  - valid date format 6-11, 7-6
  - valid number format 6-11, 7-6
- DML
  - See &SQL
- Documenting reports B-8
- Down 5-5
- Down/Across 5-5
- Drop
  - one report via action menu 3-7
  - several reports via a SQL script B-4
- DUMPREP command
  - for transforming reports 2-10

## E

- Editing commands
  - deleting text 1-15
  - pasting text 1-15
  - reading text from a file 1-16
  - writing text to a file 1-16
  - See *also* Appendix A
- Editing text objects
  - status indicator 8-8
  - to customize your report 8-3
- Elevators 1-11
- Error
  - abnormal condition 1-65
  - help 1-13
  - recovery 1-13
- Establishing security B-4

Examples  
of calls from 3GLs to SQL\*ReportWriter  
D-3 to D-6  
of a user exit E-4 to E-6  
of variable fields in text 6-18  
of wrap and variable fields in text 8-14  
of wrap fields in text 6-17

## EXEC

IAF GET E-7  
IAF PUT E-8  
ORACLE statements E-7  
SQL statements E-6  
Executing reports 3-8  
Exiting SQL\*ReportWriter, Quit 3-10  
Exporting files, using DUMPREP 2-10

## F

Fetch A-6  
locating text objects with query 8-5  
Field highlighting 5-12  
Field Label setting 6-8  
Field Name setting 6-4  
Field objects  
deleting 6-2  
inserting 6-2  
moving 6-3  
renaming 6-3  
using in headers and footers 8-13  
Field Screens 6-1  
align 6-15  
datatype 6-10  
display format 6-11  
field label 6-8  
field name 6-4  
field width 6-10  
function 6-20  
group 6-8  
lines before 6-14  
relative position 6-13  
repeat 6-19  
reset group 6-21  
skip 6-19  
source column 6-4  
spaces before 6-14

Field setting 7-4  
Field Spacing setting 5-9  
Field Width setting 6-10  
Fields Across setting 5-9  
File transfer, using DUMPREP 2-10  
File-searching method B-6  
First function  
for computed fields 6-20  
for summaries 7-5  
First line A-2  
Footer  
definition of 8-2  
referencing fields in 8-13  
Form feeds H-4  
Frequency setting 8-12  
Function setting 6-20  
%Total 6-20  
Avg 6-20  
Count 6-20  
First 6-20  
for fields 6-20  
for summaries 7-5  
Last 6-20  
Max 6-20  
Min 6-20  
R\_%Total 6-20  
R\_avg 6-20  
R\_Count 6-20  
R\_max 6-20  
R\_min 6-20  
R\_Sum 6-20  
Sum 6-20

## G

Generating reports  
from the command line 2-11  
using SQLREP 3-9  
GENREP command 2-11  
GENXTB  
form E-9  
program E-9  
Global Keys A-6  
Granting report access  
from user to user 9-4  
Group Name setting 5-4

- Group objects
  - definition of 5-1
  - deleting 5-2
  - inserting 5-2
  - moving 5-3
  - ordering 5-3
  - renaming 5-3
- Group Screens 5-1
  - field highlight 5-12
  - field spacing 5-9
  - fields across 5-9
  - group name 5-4
  - highlight 5-12
  - label highlight 5-12
  - label position 5-11
  - lines before 5-8
  - matrix 5-6
  - multi-panel 5-10
  - page break 5-6
  - print direction 5-5
  - query 5-5
  - record spacing 5-8
  - relative position 5-7
  - spaces before 5-8
- Group setting
  - for fields 6-8
- Group text objects
  - how to query 8-5
  - justification 8-11

## H

- Header
  - column 8-2
  - referencing fields in 8-13
- Help A-6
- Help menu 1-12
- Help system 1-11
  - error recovery 1-13
  - help menu 1-12
  - key help 1-12
  - keywords 1-12
  - navigation 1-12
- Hierarchy, ordering groups 5-3

- Highlight setting 5-12
  - field 5-12
  - label 5-12
- Highlighting
  - conditional highlighting of fields E-12
- Highlighting text 8-5
  - embedding printer codes in text 8-6
  - passing printer codes from user exits E-12
  - using List of values 8-5
- Hint Line 1-8
- Host
  - See Shell to host

## I

- IAPXTB
  - source file E-9
  - table E-9
- Importing files 2-12
- Input Format for parameters 10-11
- INSERT
  - See &SQL
- Insert line A-2
- Insert record above A-4
- Insert record below A-4
- Insert/replace A-3
- Installing a userid B-3
- International
  - currency symbol 10-10
  - decimal setting 10-10
  - thousands symbol 10-10

## K

- Key definitions A-1
- Key help 1-12, A-6

## L

- Label
  - highlighting 5-12
  - position setting for groups 5-11
  - setting for parameters 10-8
- Language
  - for RUNREP 2-7
  - for SQLREP 1-5
- Last function
  - for computed fields 6-20
  - for summaries 7-5
- Left
  - field alignment 6-15
  - justification for text 8-11
  - label position for fields 5-11, 6-15
- Left Margin setting 9-3
- Lexical parameter 4-7
  - creating 10-4
  - definition of 10-2
  - examples in SQL clauses 10-4
- Lines Before setting 8-9
  - for fields 6-14
  - for groups 5-8
- Linking a user exit E-9
- List of values 1-10
  - to insert columns into SELECT Statements 1-10, 4-5
  - to view possible tables 1-10
- LOADREP command 2-12
- LOGFILE
  - for RUNREP 2-6
  - for SQLREP 1-5
- Logon procedure
  - examples 1-2
- LONG datatype 6-10

## M

- Mail using RUNREP
  - sending a report to Oracle\*Mail users 2-5
- Mail using SQLREP
  - preparing to send B-4
  - report output 10-9
  - sending a report to Oracle\*Mail users 1-4
- Main menu 1-7
- Margin 8-9
- Mark A-3
- Matrix reports
  - child columns example 4-9
  - parent 2 query 4-8
- Matrix setting 5-6
- Max function
  - for computed fields 6-20
  - for summaries
- Memory and disk space requirements B-5
- Menu A-5
- Menu line 1-7
  - action 1-7
  - field 1-8
  - group 1-8
  - help 1-8
  - parameter 1-8
  - query 1-8
  - report 1-8
  - summary 1-8
  - text 1-8
- Min function
  - for computed fields 6-20
  - for summaries 7-5
- MODIFY, column in the IAPXTB table E-10
- MOVEREP command
  - chart for converting reports B-2
  - conversion program 2-13
  - convert reports from V1.0 to V1.1 B-2
- Multi-panel setting 5-10

## N

- NAME, column in the IAPXTB table E-10
- Navigation commands
  - See Appendix A
- Navigation keys A-4
- New report definition 3-3
- Next function key
  - choice A-5
  - field A-4
  - line A-2
  - record A-4
  - word A-2
- NUM\_PAGES system variable 8-12
- Number formats 6-12

## O

- Object setting
  - page 8-7
  - report 8-7
- Object setting for text 8-7
- objects 1-13
  - deleting 1-13
  - inserting 1-13
  - moving 1-14
  - renaming 1-14
  - undefined 1-14
  - undeleting 1-14
- On-line help
  - See Help system
- Open report 3-4
- Option fields 1-10
- Oracle\*Mail
  - preparing to send reports B-4
  - sending reports B-4
- output
  - output format for parameters 10-11
  - printing 2-5
  - viewing 2-9

## P

- Page
  - for text 8-7
- Page Break setting 5-6
- Page Height setting 9-2
- Page summaries 7-4
- PAGE system variable 8-12
- Page totals 7-4
- Page Width setting 9-2
- Pages, limiting output B-5
- Panel 6-13
- Panel Number setting 8-12
- PARAM
  - for RUNREP 2-7
  - for SQLREP 1-5
- Parameter Name setting 10-7
- Parameter objects 10-2
  - bind parameter 4-7
  - bind parameter examples in SQL clauses 10-3
  - created in SQL statement 4-6
  - deleting 10-5
  - in SQL statements 10-2
  - lexical parameter 4-7
  - lexical parameter examples in SQL clauses 10-4
  - moving 10-6
  - renaming 10-5
- Parameter Screen 10-1
  - copies 10-9
  - currency 10-10
  - datatype 10-7
  - decimal 10-10
  - default value 10-8
  - desformat 10-9
  - desname 10-9
  - destype 10-9
  - label 10-8
  - parameter name 10-7
  - thousands 10-10
  - width 10-8
- Parameters in SQL statements 10-2
- PARAMFORM
  - for RUNREP 2-4
  - for SQLREP 1-3

- Parent 1 Columns setting 4-9
- Parent 2 Columns setting 4-9
- Parent Query 1 setting 4-8
- Parent Query 2 setting 4-8
- Parent-child relationships 4-8
  - child Columns 4-9
  - parent 1 columns 4-9
  - parent 2 columns 4-9
- Passing arguments to a user exit E-10
- Pasting text 1-15, A-3
- Percent function
  - for computed fields 6-20
  - for summaries 7-5
- Periodic functions
  - for fields 6-20
  - for summaries 7-5
- Positioning
  - fields across 5-9
  - summaries 7-3
- Positioning summaries
  - relative position of fields 6-13
- Pre-packaged user exits E-14
  - RWECPF E-14
  - RWEGSC E-15
  - RWEIF E-15
  - RWENOP E-16
  - RWERTH E-16
- Preparing for sending reports via Oracle\*Mail B-1
- Previous function key
  - choice A-5
  - field A-4
  - line A-2
  - record A-4
  - word A-2
- Print Direction setting 5-5
- Print Group setting
  - for summaries 7-7
- Print page A-7
- Print report
  - to a file A-7
  - using system parameters 10-8
  - See also dfltrep.log
- Print screen A-7

- PRINTDEF, printer definition program
  - how to modify H-3
  - how to Use 2-12
  - purpose of H-2
  - syntax 2-12
  - See also Printer definition
- Printer codes
  - how to use in SQL\*ReportWriter 8-5
- Printer control codes
  - how to use in SQL\*ReportWriter 8-6
- Printer definition H-2
  - avoiding form feeds in reports H-4
  - compiling the printer definition file H-5
  - creating your own printer definition file H-3
  - default drivers H-2
  - example of H-3
  - format codes H-4
  - how to create printer definition files H-3
  - modifying your own printer definition file H-3
  - printer format codes H-4
  - running the PRINTDEF command (Syntax) 2-12, H-5
- Printer drivers
  - creating H-2
- Printer format codes H-4
- Printing reports
  - creating printer drivers H-2
  - how to define using PRINTDEF H-2
  - using RUNREP 2-5, 10-9
  - using SQLREP 1-3 to 1-4

## Q

- Query A-6
  - default objects 4-3
  - finding text objects 8-5
  - insert 1-7
- Query Name setting 4-4
- Query objects
  - default objects 4-3
  - deleting 4-2
  - inserting 4-2
  - using parameters in 10-2
  - viewing 4-3

- Query Screen 4-1
  - child columns 4-9
  - parent 1 columns 4-9
  - parent 2 columns 4-9
  - parent query 1 4-8
  - parent query 2 4-8
  - parent-child relationships 4-8
  - query name 4-4
  - SQL statement 4-5
- Query setting, for groups 5-5

## R

- R\_%Total function
  - for computed fields 6-20
  - for summaries 7-5
- R\_avg function
  - for computed fields 6-20
  - for summaries 7-5
- R\_Count function
  - for computed fields 6-20
  - for summaries 7-5
- R\_max function
  - for computed fields 6-20
  - for summaries 7-5
- R\_min function
  - for computed fields 6-20
  - for summaries 7-5
- R\_sum function
  - for computed fields 6-20
  - for summaries 7-5
- Reading text from a file 1-16
- READONLY
  - for RUNREP 2-6
- Record
  - definition of 1-9
- Record Spacing setting 5-8
- Refresh key A-6
- Relative Position setting
  - of right, below, panel (for fields) 6-13
  - right, below (for groups) 5-7
  - right, below, page, margin (for text) 8-9
- REMARKS, column in the IAPXTB table E-10
- Renaming reports 3-6
- Repeat key A-6
- Repeat on Page Overflow setting 8-11

- Repeat setting 6-19
- REPORT
  - for RUNREP 2-3
    - frequency Of 8-12
    - summary print group 7-7
    - text object of 8-7, 8-12
- Report access 9-4
- Report history
  - created by 9-5
  - created date 9-5
  - created version 9-5
- Report Screen 9-1
  - bottom margin 9-3
  - comments 9-4
  - created by 9-5
  - created date 9-5
  - created version 9-5
  - left margin 9-3
  - page height 9-2
  - page width 9-2
  - parameter form hint 9-4
  - parameter form title 9-3
  - parameter status 9-4
  - right margin 9-3
  - top margin 9-3
- Reports
  - documenting B-8
  - dropping B-4
  - on reports B-8
  - upgrading from previous versions B-2
- Reset Group setting
  - for fields 6-21
  - for summaries 7-8
  - restriction for page 4-6
- Restrictions on writing user exits E-13
- Returning values from a user exit E-n
- Right
  - field alignment 6-15
  - justification for text 8-11
  - relative position of fields 6-13
  - relative position of groups 5-7
  - relative position of text object 8-9
- Right Margin setting 9-3
- Run-time
  - files opened during B-7

- Run-time interface 2-1
  - browse mode 2-9
  - DUMPREP 2-10
  - GENREP 2-11
  - LOADREP 2-12
  - MOVEREP 2-13
  - PARAMFORM 1-3
  - PRINTDEF 2-12
  - run-time parameter form 2-8
  - RUNREP 2-3
  - TERMDEF 2-13
  - window mode 2-9
- Run-time parameter form 2-8
  - moving parameters on the form 10-6
  - parameter hint 9-4
  - parameter status 9-4
  - parameter title 9-3
- Running functions
  - for fields 6-20
  - for summaries 7-5
- RUNREP command
  - ARRAYSIZE 2-4
  - BATCH 2-7
  - BUFFERS 2-6
  - CMDFILE 2-4
  - command line syntax rules 2-7
  - COPIES 2-5
  - CURRENCY 2-5
  - DECIMAL 2-6
  - DESFORMAT 2-5
  - DESNAM 2-5
  - DESTYPE 2-5
  - LANGUAGE 2-7
  - LOGFILE 2-6
  - PARAM 2-7
  - PARAMFORM 2-4
  - preparing to send reports via Oracle\*Mail B-4
  - READONLY 2-6
  - REPORT 2-3
  - sending reports to Oracle\*Mail users 2-5
  - syntax 2-3
  - TERM 2-4
  - THOUSANDS 2-6

- RWECPF
  - pre-packaged user exit E-14
- RWEGSC
  - prepackaged user exit E-15
- RWEIF
  - prepackaged user exit E-15
- RWENOP
  - pre-packaged user exit E-16
- RWERTH
  - pre-packaged user exit E-16

## S

- Screen layout 1-7
  - hint line 1-8
  - main menu 1-7
  - menu line 1-7
  - status line 1-9
  - title line 1-8
  - work area 1-8
- Scroll A-4
  - left/right A-4
  - leftmost/rightmost A-5
  - prev/next A-4
  - top/bottom A-5
  - up/down A-4
- Search, search and replace A-3
- security
  - establishing B-4
- Select A-5
- SELECT Statement setting 4-5
- Sending report to Oracle\*Mail users
  - preparing to send B-4
  - using RUNREP 2-5
  - using SQLREP 1-4, 10-9
- Setting
  - definition of 1-9
- Sharing reports
  - See Access privileges
- Shell to host A-6
- Skip setting 6-19
- Source setting 6-4
- Spaces Before setting 8-10
  - for fields 6-14
  - for groups 5-8

- SQL commands
  - in user exits E-6
- SQL error help 1-13
- SQL naming standards 6-4
- SQL statements
  - arbitrary SQL statement 6-6
  - SELECT INTO 6-6
  - using &SQL as a field's source 6-4 to 6-6
- SQL\*ReportWriter Screens 1-9
  - field screens 6-1
  - group screens 5-1
  - help screens 1-11 to 1-13
  - parameter screen 10-1
  - query screen 4-1
  - report screen 9-1
  - summary screens 7-1
  - text screen 8-1
- SQLREP command
  - ARRAYSIZE 1-3
  - BUFFERS 1-5
  - CMDFILE 1-3
  - Command line syntax rules 1-5
  - COPIES 1-4
  - CURRENCY 1-4
  - DECIMAL 1-5
  - DESFORMAT 1-4
  - DESNM 1-4
  - DESTYPE 1-3
  - LANGUAGE 1-5
  - LOGFILE 1-5
  - PARAM 1-5
  - PARAMFORM 1-3
  - preparing to send reports via Oracle\*Mail B-4
  - sending reports to Oracle\*Mail users 1-4
    - syntax 1-2
  - TERM 1-3
  - THOUSANDs 1-4
  - USERID 1-2
- SRW132, report on reports B-8
- SRW175, report on reports B-8
- SRW80, report on reports B-8
- Status indicator
  - on text screen 8-8
- Status line 1-9
- Subfooter 8-2
- Sum function
  - for computed fields 6-20
  - for summaries 7-5
- Summary Name setting 7-4
- Summary objects
  - creating 7-2
  - creating multiple summaries 7-3
  - deleting 7-2
  - moving 7-2
  - positioning 7-3
  - renaming 7-2
- summary screens 7-1
  - datatype 7-6
  - display format 7-6
  - field 7-4
  - function 7-5
  - print group 7-7
  - reset group 7-8
  - summary name 7-4
  - width 7-6
- System parameters 10-8 to 10-10
- System variables 8-12
  - &DATE 8-12
  - &NUM\_PAGES 8-12
  - &PAGE 8-12
  - as field sources 6-5
  - as fields 6-4
  - how to change display format of 6-5
- System-owned SQL\*ReportWriter tables
  - definition of C-1

## T

### TERM

- for RUNREP 2-4

- for SQLREP 1-3

TERMDEF, terminal definition program  
2-13, G-1

Terminal definition 2-13, G1 to G15

- changing an existing keymapping G-3
- consistent with SQL\*Forms, SQL\*Menu G-16
- creating a private keymapping G-2
- cursor movement options G-15
- highlight codes listing G-12
- highlighting codes G-9
- key definition file G-4
- sample terminal definitions file entry G-10
- supporting a new terminal G-3
- terminal capability codes G-12 to G-13
- terminal definitions file (TERMDEF) G-5

### Terminals

- See Terminal definition

### Text

- copying 1-15
- creating modifying suppressing panels  
8-3 to 8-4
- deleting 1-15
- highlighting 8-5
- manipulating 8-3
- pasting 1-15
- reading text from a file 1-16
- writing text to a file 1-16

Text editing keys A-2

### Text objects

- body 8-2
- column heading 8-2
- examples of wrap and variable fields in 8-14
- footer 8-2
- header 8-2
- subfooters 8-2
- title page 8-2
- trailer page 8-2

Text panels 8-3 to 8-4

- creating 8-4
- customizing 8-3
- suppress printing 8-3

Text Screen 8-1

- frequency 8-12
- justification 8-11
- lines before 8-9
- object 8-7
- panel number 8-12
- query by example 8-5
- relative position 8-9
- repeat on page overflow 8-11
- spaces before 8-10
- status 8-8
- text 8-12
- text type 8-8

Text setting 8-12

Text Type setting 8-8

- body 8-8
- column heading 8-8
- footer 8-8
- header 8-8
- title page 8-8
- trailer page 8-8

### THOUSANDS

- for RUNREP 2-6
- for SQLREP 1-4
- system parameter 10-8, 10-10

Thousands setting 10-10

Title line 1-8

Title page 8-2

Top Margin setting 9-3

Trailer page 8-2

TYPE, column in the IAPXTB table E-10

Types of user exits E-2

- non-Oracle E-3
- OCI (Oracle Call Interface) E-3
- ORACLE precompiled E-3

## U

- Undefined objects 1-14, 4-3
- Undelete line A-2
- Undelete record A-4
- Undelete word A-2
- Undo A-6
- UPDATE
  - see &SQL
- Upgrading reports from previous versions B-2
- user exits
  - as field sources 6-5
  - building a function E-4
  - general steps E-4
  - linking E-9
  - passing arguments to a user exit E-10
  - pre-packaged with SQL\*ReportWriter E-14
  - restrictions E-13
  - returning values from a user exit E-11
  - types E-2
  - uses E-2
  - what they are E-2
- User Interface 1-1
  - alert box 1-11
  - dialog box 1-11
  - help system 1-11
  - list of values 1-10
  - menus 1-10
  - option fields 1-10
  - screen layout 1-7
  - SQLREP 1-2
- User-owned SQL\*ReportWriter tables
  - definition of C-1
- USERID
  - for RUNREP 2-4
  - for SQLREP 1-2
- Userid, installing a B-3

## V

- Variable fields
  - examples of variable fields 6-18, 8-14 to 8-15
  - field alignment 6-16
  - rules for 6-16 to 6-17
  - rules in text objects 8-13 to 8-14

## W

- Width setting
  - for fields 6-10
  - for parameters 10-8
  - for summaries 7-6
  - for text 8-10
- Window A-5
- Window mode 2-9
- Work area 1-8
- Wrap fields 6-17
  - examples of wrap fields 6-17 to 6-18
  - field alignment 6-16
  - rules for 6-16 to 6-17
  - rules in text objects 8-13 to 8-14
- Write file A-3
- Writing text to a file 1-16