CA Ideal™ for CA Datacom®

Generating Reports Guide

Version 14.02

CA Technologies Product References

This document references the following CA products:

- CA Datacom®/DB
- CA eMail+
- CA Ideal™ for Datacom® (CA Ideal)
- CA Ideal™ for DB2
- CA Ideal™ for VSAM
- CA CICSORT™
- CA Sort®

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Introduction 9

Introduction to RDF 9
Report Layout 9
    RDF Prompt Screens 10
    Accessing RDF 12
PF Key Assignments 12
Setting Report Options for the Session 14
Basic Steps to Create a Report 16
    Simple Sample Report 17
    Report Maintenance Menu 19
Creating a Report Definition from a Dataview 20

Chapter 2: Generating Reports 21

Specifying General Page Format 21
    Modifying the Format for One Report 21
Defining the Report Data 23
    Using a Dataview 23
    Including Other Types of Fields 25
    Including Fields from Multiple Dataviews 28
    Arranging the Report Data 29
    Placing Individual Fields in the Detail Line 29
    Printing Multiple Lines with One Detail 32
    Grouping Details for Individual Printing 33
Sequencing the Report Data 37
    Sorting Requirements 38
Formatting Based on Column Value Control Break 40
    Identifying Control Breaks on the Report 42
Headings and Footings 45
    Defining Report and Page Headings and Footings 45
    Defining Control Break Headings and Footings 52
    Defining Column Headings 55
Generating the Bottom Line 62
    Identifying the Function 63
    Obtaining Multiple Values 64
    Including Report Functions in Footings 66
    Calculating Values in a Footing 70

Printing Report Summaries 71
Generating Multiple Reports 72
RELEASE REPORT 73
Changing Output Destination 73
Changing Destination for the Application Run 73
Changing the Output File Dynamically 74
Assigning Other Options 76
Reporting Exception Conditions 76
Using <<EMPTY>> 77
Using <<ABORT>> 78

Chapter 3: Checking the Report Definition 79

Displaying the Report Definition 79
Printing the Report Definition 79
Prototyping the Report 80
Prototyping Prerequisites 80
PRODUCE Command 81

Chapter 4: Generating Reports Procedurally 83

Listing All Rows 83
Creating Page Breaks 83
Controlling Page Breaks in a Report 84
Listing Several Reports in One Application Run 84
Formatting the Report 84
Formatting Procedurally 85
Formatting with Panels 86
Printing and Prototyping Reports 87
Providing Summary Information 89
Listing Several Reports on One Page 89
Controlling the Report Destination 90
Using LIST Instead of RDF 91

Appendix A: Fill-In Descriptions 93

Report Identification Fillin 94
Report Parameters Fillin 95
Heading/Footing Definition Fillin 105
Detail Definition Fillin 112
Report Groups 113
PrimaryGroup 113
SecondaryGroup 113

Displaying the Detail FillIn 113
Column Headings Definition Fillin 123

Appendix B: Sample Reports 125

Sample Report 1 126
Sample Report 2 129
Mailing Labels 131
Parameter Specifications 132
Using a Table Defined in Working Data 133
Detail Definition Provides Report Definition 133
One PRODUCE, Multiple Rows 134

Chapter 1: Introduction

The Report Definition Facility (RDF) automates the process of generating reports. A CA Ideal program requires only a PRODUCE statement to generate the output. RDF controls the report format, including headings, footings, and data format.

When you use RDF to generate a report, the PRODUCE statement in your program passes the data to RDF, which writes the data to an output file named for the report. CA Ideal specifically uses the data for the report. If a DD statement did not specify a filename, AUXPRINT is used. This file accumulates the output as the program progresses. When the program terminates or the report is released, CA Ideal takes the output data and, based on the RDF specifications, formats the report. This formatting includes sorting (in z/OS and VSE batch only), pagination, annotated totals, column headings, and so forth. The completed report is automatically sent to the printer.

Introduction to RDF

In CA Ideal, the functions of defining the report and generating the output data are distinct. This lets the programmer focus on solving the business problem while RDF controls how the solution is presented. Report maintenance is accomplished by modifying the report definition using RDF. The application procedure code normally does not require any modification.

Report Layout

Most reports consist of several parts, including the following:

Component	**Includes**
Title Page	Report heading
Body	Page headings Page numbers Report date Column headings Details (the data) Controls break footings. Logical grouping of the data based on the value in a field. Can include annotated values.

Component	Includes
Summary	Final information including: Summary Title and report totals Report footing

You must also define the page layout. The page layout includes the page width, length, and spacing between the parts of the report.

RDF Prompt Screens

Using RDF, you define the various report components and attributes through a series of fill-in screens. The RDF fill-in screens define the report characteristics as follows:

Identification

Provides descriptive information about the report. This information establishes the relationship in the dictionary facility between the report and any programs using the report.

Parameters

Defines global attributes for the report, including:

- Lines-per-page
- Report width
- Page numbering
- Line spacing
- Date format
- Simple page heading

Heading

Defines detailed report and page headings and footings.

Detail

Defines specifications for the report detail lines, including:

- Individual field placement
- Sorting
- Control break report functions
- Edit patterns

Column Headings

Defines specialized column headings

When a new report is created, the Identification fill-in automatically displays. After that initial display, each fill-in screen, including the Identification fill-in, must be accessed directly using a command or a PF key. The same screens modify existing reports.

Accessing RDF

To access RDF to create, edit, or display a report definition, you can select the appropriate option from the Report Maintenance Menu or enter the CREATE, EDIT, or DISPLAY REPORT command. You can control the fill-in displayed when you enter the command with options in the command, or you can accept the default.

- For CREATE, the default is the Identification fill-in.
- For EDIT and DISPLAY, the default is the Detail fill-in.

The functions that CA Ideal provides to define and maintain report definitions are presented in the Report Maintenance Menu. To access this menu, select option 4 on the Main Menu or enter the REPORT command.

```
=>
=>
=>
------------------------------------------------------------------------
 IDEAL   : REPORT MAINTENANCE     (RPT)                    SYS: DOC  MENU

 Enter desired option number ===>   There are    8  options in this menu:

 1. DISPLAY/EDIT        - Display or edit a report definition
 2. CREATE              - Create a report definition
 3. PRINT               - Print a report definition
 4. DELETE              - Delete a report definition
 5. DUPLICATE           - Duplicate report to new name
 6. DISPLAY INDEX       - Display index of report names in system
 7. PRINT INDEX         - Print index of report names in system
 8. PRODUCE             - Produce a report facsimile
```

To return to the menu when you complete RDF activities, press the Return key (PF2) or enter the appropriate command.

For information on accessing the fill-in screens in RDF with an explanation of each fill-in field, see the appendix Fill-in Descriptions.

PF Key Assignments

When you are using the Report Definition Facility, the PF keys are assigned the following commands.

HELP (PF1)

Displays a panel or series of panels that contain information explaining how to complete the current function.

RETURN (PF2)

Returns from a help panel to the program component display or from the program to the menu that selects the program.

PRINT SCREEN (PF3)

Generates a hardcopy printout of the current screen contents.

PARAMETERS (PF4)

Displays the report parameters. If you are editing the report definition when you enter this command, you can edit the displayed report parameters.

HEADING (PF5)

Displays the page heading definition. If you are editing the report definition when you enter this command, you can edit the displayed page heading definition.

DETAIL/COLUMN [HEADINGS] (PF6)

Displays the report detail definition. If the report detail definition is displayed, this PF key displays the column headings definition. If the column headings definition is displayed, this PF key displays the report detail definition. You can edit the displayed definition if you are editing the report definition when you enter this command.

SCROLL BACKWARD (PF7)

Displays the previous frame in the current component.

SCROLL FORWARD (PF8)

Displays the next frame in the current component.

FIND (PF9)

Repeat a previous FIND string command for report detail or heading data.

SCROLL TOP (PF10)

Positions to the first line of the component.

SCROLL BOTTOM (PF11)

Positions to the bottom of the component.

INPUT (PF12)

Opens a window of null lines preceding the first line of the component or at the current cursor position. Unused null lines in the window are deleted when you press the Enter key after INPUT.

Setting Report Options for the Session

To change the default values for report parameters on report definitions to create later in the current session, you can issue SET REPORT commands. You can issue these commands at any time before you create the report. They are then in effect for the remainder of the current session or until they are reset. Existing report definitions maintain the values that were in effect when the report definition was defined.

You can enter these commands into a data member called SIGNON. They then automatically execute whenever you sign on to CA Ideal. In effect, they are your defaults across sessions.

The following are brief descriptions of the SET REPORT commands. For more information on any of these commands, see the *Command Reference Guide*.

SET REPORT LINES

Specifies the total number of lines for each printed page of the report.

SET REPORT WIDTH

Specifies the number of characters per line of the report.

SET REPORT SPACING

Specifies the default number of blank lines between printed detail lines.

SET REPORT GAP

Specifies the default number of blank characters to be left between defined columns of detail lines.

SET REPORT CONTHEAD

Specifies whether headings print when a control break occurs. A control break is a change in the value of a control field.

SET REPORT CONTFOOT

Specifies whether footings print when a control break occurs.

SET REPORT DATEFOR 'date-pattern'

Specifies the default format of the date printed on reports.

You can use any logical combination of the following date components to specify a date format.

The specifications in the following table give the corresponding results when a report is produced, assuming that at the time of production the date is January 14, 2006.

Component Notation	**Meaning**	**Example assuming (January 14, 2006)**
YEAR	Year in full	2006
YY	Year without century	06
Y	Year without decade	6
MONTH	Month spelled out (uppercase)	JANUARY
LCMONTH	Month spelled out (initial letter uppercase)	January
MON	Month abbreviation (uppercase)	JAN
LCMON	Month abbreviation (initial letter uppercase)	Jan
MM	Month number, with leading zero if necessary	01
M	Month number with no leading zero	1
DD	Day with leading zero if necessary	14
D	Day with no leading zero	14
DDD	Julian day, numeric day of the year (1-366)	014
WEEKDAY	Day spelled out (uppercase)	SUNDAY
LCWEEKDAY	Day spelled out (initial letter	Sunday
DAY	Day abbreviation	SUN
LCDAY	Day abbreviation (initial letter uppercase)	Sun

Any characters except uppercase alphabetic in the date pattern remain unchanged.

The site administrator defines the actual text indicated by the keywords MONTH, LCMONTH, MON, LCMON, WEEKDAY, LCWEEKDAY, DAY, and LCDAY for each site in the file DATETBL.IDL. For more information about defining this text, see the *Working in the Environment Guide*.

SET REPORT DATEPOS

Specifies the default position where the date displays in the heading or footing.

SET REPORT NULLSYM

Specifies the default symbol that represents null values on the report.

SET REPORT PAGEFMT

Specifies the default page number format.

SET REPORT PAGEPOS

Specifies the default position where the page number displays in the heading or footing.

Basic Steps to Create a Report

The following are the minimum steps necessary to create a report using RDF:

1. Define the report.

   Enter the CREATE REPORT command to begin the process. The Identification fill-in displays automatically when the CREATE REPORT command is issued. When you assign a name to a report in the CREATE REPORT command or in the Identification fill-in, the report is formally identified in the dictionary facility.

   Enter the command DETAIL to access the Detail fill-in. Define the report fields on the Detail fill-in.

2. Identify the report to the program.

   Add the report to the program resources. A program does not compile if it attempts to generate a report that is not specified as a resource.

   Any variables or dataviews the report uses must also be defined as resources.

3. Code the PRODUCE statement.

   Include the PRODUCE statement in the program to generate the report output.

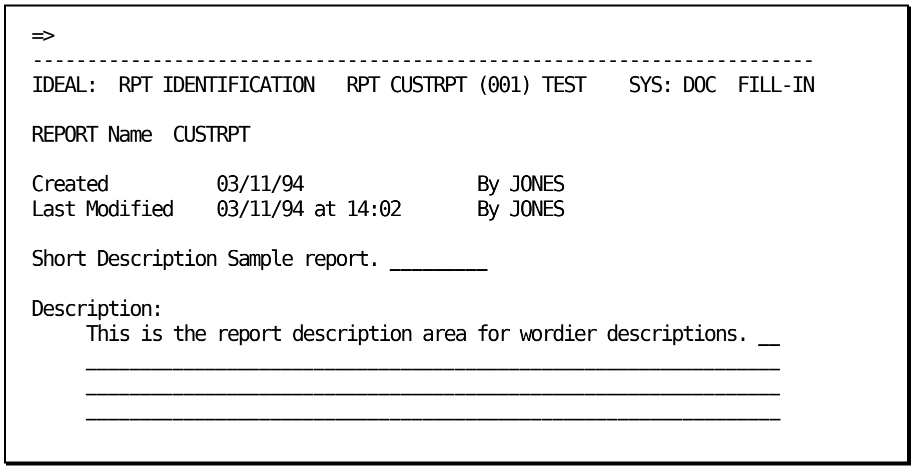
Simple Sample Report

Using the three steps listed previously, you can create a program that prints a report containing the names of all of the customers in Texas.

1. Define the report named CUSTRPT, starting with the following command:

   ```
   CREATE REPORT CUSTRPT
   ```

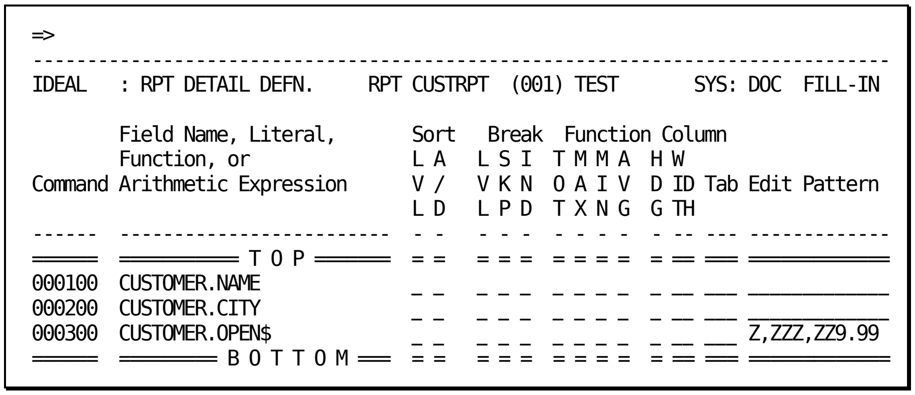
   The Identification fill-in screen displays.



```
=>
-----------------------------------------------------------------------
IDEAL:  RPT IDENTIFICATION   RPT CUSTRPT (001) TEST    SYS: DOC  FILL-IN

REPORT Name  CUSTRPT

Created         03/11/94               By JONES
Last Modified   03/11/94 at 14:02      By JONES

Short Description Sample report. _________

Description:
     This is the report description area for wordier descriptions. __
     ______________________________________________________________
     ______________________________________________________________
     ______________________________________________________________
```

The identification information shows the name assigned to the report, the creation date, and the user ID assigned to the report creator. The Last Modified information contains the creation date until the report is edited. A Short Description is included. You can also include a longer description, up to five lines.

After the Identification fill-in is complete, the Detail fill-in is accessed to define the content. Each field must be listed in order, by name. You can specify the format or edit pattern for each field with several other attributes.



```
=>
------------------------------------------------------------------------------
IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST      SYS: DOC  FILL-IN

        Field Name, Literal,      Sort   Break  Function Column
        Function, or              L A   L S I  T M M A  H W
Command Arithmetic Expression     V /   V K N  O A I V  D ID Tab Edit Pattern
                                  L D   L P D  T X N G  G TH
------  ------------------------  - -   - - -  - - - -  - -- --- ------------
======  =========== T O P ======  = =   = = =  = = = =  = == === ============
000100  CUSTOMER.NAME             _ _   _ _ _  _ _ _ _  _ __ ___ ____________
000200  CUSTOMER.CITY             _ _   _ _ _  _ _ _ _  _ __ ___ ____________
000300  CUSTOMER.OPEN$            _ _   _ _ _  _ _ _ _  _ __ ___ Z,ZZZ,ZZ9.99
======  ========= B O T T O M ==  = =   = = =  = = = =  = == === ============
```

This report contains the customer name, city, and outstanding amount owed. The report fields use data from columns in the CUSTOMER dataview, as the Field Names show: CUSTOMER.NAME, CUSTOMER.CITY, and CUSTOMER.OPEN$.

Notice that an edit pattern is specified for the field CUSTOMER.OPEN$, which is a numeric field. If you do not specify an edit pattern, the edit pattern as defined for the column in the Datadictionary is used. For numeric data, if an edit pattern was not defined in the Datadictionary, a default pattern of Z(*i*).Z(*d*) is used, where *i* is the number of integers to the left of the decimal point and *d* is the number of decimal places to the right of the decimal point. For information on valid edit patterns, see the appendix "Fill-in Descriptions."

2. Identify the Report to the Program.

   The following report must be identified to the program on the Resources fill-in:

```
=>
=>
=>
IDEAL : Resource Edit Panel   PGM SAMPLE (001) TEST         SYS: DOC   DISPLAY

 Command Type    Name of DVW/PGM/PNL or RPT        Version  System  Qual?
 ====== ===     ========== T O P =============     ====    ===     =
 000001 RPT     CUSTRPT                             001     DOC
 ====== ===     ======= B O T T O M ==========     ====    ===     =
```

3. Code the PRODUCE Statement.

   You can code the PRODUCE statement anywhere in the program. Usually, it is coded in a LOOP or FOR construct to generate the report details. In this example, the PRODUCE statement is coded in the FOR EACH construct. The WHERE clause criteria limit the set to all customers in Texas.

```
FOR EACH CUSTOMER
    WHERE STATE EQ 'TX'
    PRODUCE CUSTRPT
ENDFOR
```

   With each iteration of the FOR construct, the PRODUCE statement writes the contents of the specified columns of the current row to the report file as a detail line. A MOVE or SET statement is not needed. The customers are listed in the sequence in which the database accessed them, therefore, they are not in alphabetical order by name or city.

   There is nothing in the program that formats the data or deals with pagination, headings, and so on.

The following is a sample of the resulting report:

```
NAME                          CITY                  OPEN$
CHEMICAL MUTUAL               FORT WORTH           931.72
GULF LAND USA                 DALLAS             7,100.00
TEXAS LIFE & CASUALTY CO.     DALLAS               543.21
AFTON INDUSTRIES              DALLAS             1,234.51
PALMOLIVE INNS                FORT WORTH           758.93
  .
  .
  .
```

In this example, the column headings and width were taken from the definition in the dictionary facility. By default, two spaces are inserted between each column. One line is inserted between the column heading and the detail lines. There are no other headings or footings in this report.

Report Maintenance Menu

The functions CA Ideal provides to define and maintain report definitions are presented in the Report Maintenance Menu. To access this menu, select option 4 on the Main Menu or enter the REPORT command.

```
=>
=>
=>
------------------------------------------------------------------------
   IDEAL   : REPORT MAINTENANCE     (RPT)                    SYS: DOC  MENU

   Enter desired option number ===>   There are    8  options in this menu:

   1. DISPLAY/EDIT         - Display or edit a report definition
   2. CREATE               - Create a report definition
   3. PRINT                - Print a report definition
   4. DELETE               - Delete a report definition
   5. MARK STATUS          - Mark report status to production or history
   6. DUPLICATE            - Duplicate report to new name
   7. DISPLAY INDEX        - Display index of report names in system
   8. PRODUCE              - Produce a report facsimile
```

Each option on the Report Maintenance Menu is described in the following pages. The order in which the options are presented is their logical order of use, but it does not reflect a required sequence for building a report definition. Each option shows the following:

- The *command syntax* equivalent to the menu selection. For more information on the command syntax, see the *Command Reference Guide*.
- Any fill-in required by the selected function.

If the particular fill-in or portion of a fill-in was already completed, the information displays as it was last entered. If the fill-in was never completed, a fill-in that contains default values, where appropriate, displays.

After a fill-in is completed, press any transaction key to apply the modified data. The Enter key always applies the data and displays the current fill-in.

Creating a Report Definition from a Dataview

To create a report definition, enter the command CREATE REPORT or select option 2 from the Report Maintenance Menu. You are prompted for the rest of the command. To bypass the prompter, enter the complete CREATE REPORT command. The format of the CREATE REPORT command is:

`CREATE REPORT [name [{FROM|USING} {*|DVW [`*`auth-id`*`.]`*`dvw-name`*`}]`

name

Specifies the one- to eight-character name assigned to the new report definition. This value is required only because you are entering a USING or FROM clause, such as one of the following:

`FROM *`

`FROM DVW` *`auth-id.dvw-name`*

`USING *`

`USING DVW` *`auth-id.dvw-name`*

- These parameters indicate that the report is created from the specified dataview. You can specify the dataview explicitly by entering the *dvw-name* with the authorization ID for SQL dataviews or, if a dataview definition is current, you can specify the current dataview by entering an asterisk (*). A dataview definition is current if it is the last entity you displayed or edited.
- The report definition is actually created when you access the EDIT REPORT DETAIL fill-in. A primary group, containing exactly the fields in the dataview, is automatically defined. The fields are arranged across the report page, using the spacing specified by default in the report Parameters fill-in. No report, page, or break headings are defined. Column headings are determined from the dataview definition.

When you enter the CREATE REPORT command, the report Identification fill-in displays. This fill-in establishes the report name (if it was not entered as part of the CREATE REPORT command) and provides identification information. Until the report is named in the CREATE REPORT command or on the report Identification fill-in, the new definition does not exist. Naming the report enters the report definition as an entity in the dictionary. When the name is entered on the Identification fill-in that report becomes the current report.

Chapter 2: Generating Reports

This chapter explains how to define each element of a report using RDF and includes examples of program procedures used with the report definitions to create reports. Many of the RDF fill-in screens are partially shown, focusing on the area being described. For a complete explanation of all the fields on the RDF fill-in screens, see the appendix "Fill-in Descriptions."

Specifying General Page Format

The specifications for the general page format of the report are global attributes. Defaults for such global report characteristics as the number of lines per page, page width, date format, page number placement, and headings are specified when CA Ideal is installed. You can override these defaults on a report-by-report basis using the Parameter fill-in, or on a session basis using the SET REPORT options listed in Setting Report Options for the Session in the chapter "Report Definition Facility." You can display the values currently in effect by entering the DISPLAY SESSION REPORT command. For more information about specifications for the general page format, see the *Command* Reference *Guide*.

Modifying the Format for One Report

Although you can generate a report using default assignments for the layout, you can modify these values using the Parameter fill-in to suit each report.

The Parameter fill-in prompts for attributes that apply globally to the report. These attributes include:

- The length and width of a page
- The spacing between lines and columns
- Column headings and how they are highlighted
- Control breaks
- Group continuation indication
- Heading definition
- Summary information
- Date and page specification

You can override many of the global attributes on the Parameter fill-in with values specified on other fill-ins. The Detail fill-in specifies attributes for individual columns. The Report Header fill-in and Column Header fill-in specify individual headings.

You can modify the defaults on the Parameter fill-in by overtyping the appropriate fields. The range of valid values for each attribute displays next to the data entry field. The following segment of the Parameter fill-in shows the first five fields. These fields define the page length, width, and the number of spaces between lines and columns and between headings and details.

```
=>
 ---------------------------------------------------------------- Partially
shown
 IDEAL   : RPT PARAMETERS       RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

 Lines per page on printout          060  (0 thru 250)
 Report width                        132  (40 thru 230)
 Spacing between lines               1    (1 thru 3)
 Spacing between columns             02   (0 thru 66 OR A=Automatic)
 Spacing after page and column hdgs  1    (0 thru 9)
    .
    .
    .
```

The lines-per-page value specifies the total number of lines on the printed page. This includes all headings and footings, spacing, and details. You can suppress page breaks by specifying a 0 for lines-per-page. The report then prints on every line including any perforation lines.

When page breaks are not suppressed, the value specified by lines-per-page represents the maximum number of lines on the page, including all headings, footings, details, and blank lines. To determine whether a page break is to occur, RDF first determines the number of lines required for the page heading, column headings, page footing, and associated spacing. This value is subtracted from the lines-per-page specification. The remaining lines are available for the page body consisting of the details and control break headings and footings.

When a detail that does not cause a control break is produced, a page break occurs if the complete detail does not fit on the page. When a detail that causes a control break is produced, a page break occurs if the appropriate control break footing or heading does not fit on the page.

Other global attributes specified on the Parameter fill-in are described under relevant topics. For example, the Parameter fill-in area for specifying a page heading is described in the section on headings.

For detailed information about every field on the Parameter fill-in, see the appendix "Fill-in Descriptions."

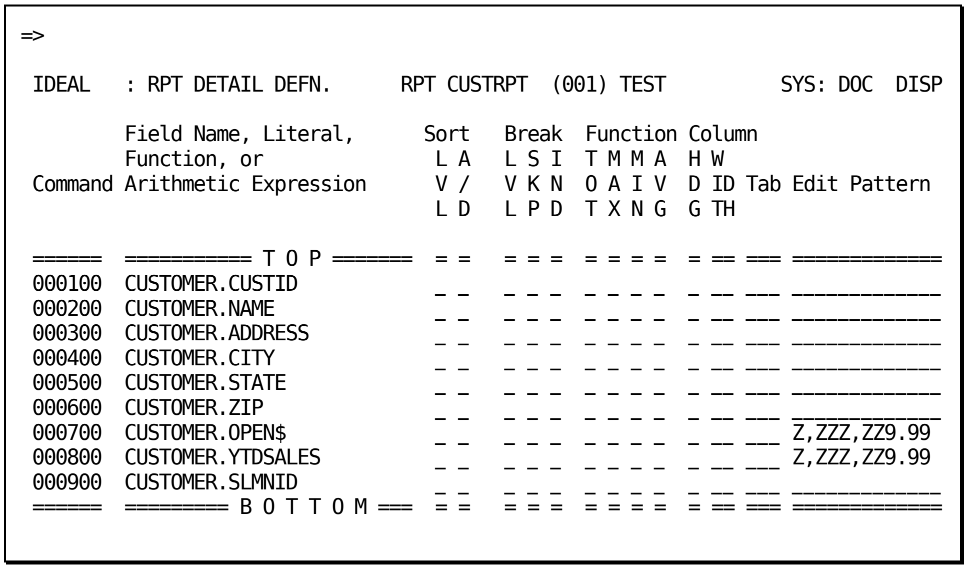
Defining the Report Data

The Detail fill-in defines the actual data or body of the report. Fields are listed across the report row in the order in which they are listed on the Detail fill-in. A complex report, including column headings, column totals, control breaks, arithmetic functions, and some formatting, can be generated from the information on this screen alone.

Using a Dataview

Frequently, the details for a report are based on one or more columns in a dataview. When a report is created based on a dataview, the report creator needs to enter the column names on the Detail Definition fill-in screen. In most cases, you need to prefix the column name with the dataview name.

The following screen shows an example of the column names of the CUSTOMER dataview specified on the Detail fill-in. The column names were used as field names and, by default, display as headers for the report columns. Edit patterns were assigned to the CUSTOMER.OPEN$ and CUSTOMER.YTDSALES fields.


```
=>

 IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH

 ======  =========== T O P ======   = =   = = =  = = = =  = == === ============
 000100  CUSTOMER.CUSTID            _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000200  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000300  CUSTOMER.ADDRESS           _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000400  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000500  CUSTOMER.STATE             _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000600  CUSTOMER.ZIP               _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000700  CUSTOMER.OPEN$             _ _   _ _ _  _ _ _ _  _ __ ___ Z,ZZZ,ZZ9.99
 000800  CUSTOMER.YTDSALES          _ _   _ _ _  _ _ _ _  _ __ ___ Z,ZZZ,ZZ9.99
 000900  CUSTOMER.SLMNID            _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === ============
```

If the length of the data exceeds the specified edit pattern, it is truncated. Null values in fields specified with null indicators appear as question marks padded with blanks to the total field length. Truncation, padding, and justification in these situations are indicated in the following table.

Value	Alphanumeric Data	Numeric Data
Exceeds specified edit pattern Value is...	Truncated on right	Truncated on left, indicated by leading "?"
Null with indicator Question marks are...	Left justified, followed by blanks	Right justified, preceded by blanks

Controlling Wrap

The report shown in the following sample was created by including all columns of the dataview in the Detail fill-in, using the current defaults for any other attributes.

```
CUSTID  NAME                 ADDRESS                 CITY        STATE ZIP

OPEN$        YTDSALES     SLMNID

B-0230  CHEMICAL MUTUAL      555 MAIN STREET         FORT WORTH  TX    761026102
      931.72     7,250.00      00-725
B-1450  UNION TRANSPORTATION 367 WEST SPRING STREET  GALVESTON   TX    775507550
    1,807.30       450.00      27-745
 .
 .
 .
```

When the length of the data or the length of the headings exceeds the defined line length, the output wraps to the next line. The compiler issues a warning message indicating the wrap condition, but the program can still run. You can control the wrapping by:

- Changing the specification for the length of the line on the Parameter fill-in (which might not resolve the wrapping).
- Eliminating columns from the report to include only those that fit on the line.
- Specifying the individual field format.

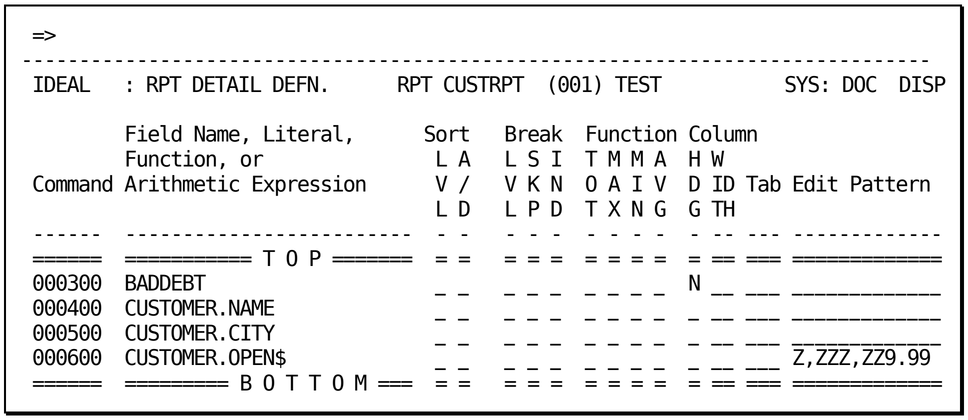
Including Other Types of Fields

You do not need to specify all fields from the dataview and, similarly, you can add new fields that are not in a dataview. The data for these fields can come from working data or parameter fields defined in your program, literals, expressions (including functions and mathematical expressions), and panel fields.

Parameter or Working Data Fields

To include a working data or parameter data variable in your report, specify the field name defined in the working data or parameter fill-in as the field name on the detail fill-in.

For example, in a report of customers in Texas, it might be important to note which customers are in arrears. Assume that a working data variable named BADDEBT is defined as a one-character alphanumeric value. In this program, when the OPEN$ amount is greater than zero and the ACTDT, last activity date, is greater than six months, an asterisk is assigned to BADDEBT. Otherwise, BADDEBT contains a blank. The value of BADDEBT and the customer name, city, and outstanding amount owed prints. The detail fill-in for this report is shown in the following screen:



```
=>
------------------------------------------------------------------------------
IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

        Field Name, Literal,      Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
------  ------------------------- - -   - - -  - - - -  - -- --- -------------
======  =========== T O P ======= = =   = = =  = = = =  = == === =============
000300  BADDEBT                    _ _   _ _ _  _ _ _ _  N __ ___ _____________
000400  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000500  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000600  CUSTOMER.OPEN$             _ _   _ _ _  _ _ _ _  _ __ ___ Z,ZZZ,ZZ9.99
======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

The pertinent program segment is coded as follows:

```
FOR EACH CUSTOMER
     WHERE STATE EQ 'TX'
   IF OPEN$ GT 0 AND
      $TODAY - ACTDT GT 180
        SET BADDEBT = '*'
   ELSE
        SET BADDEBT = $SPACES
   ENDIF
   PRODUCE CUSTRPT
ENDFOR
```

Although ACTDT is evaluated in the program, it is not used in the report. The output appears as shown in the following sample:

```
  NAME                          CITY          OPEN$

* CHEMICAL MUTUAL               FORT WORTH         931.72
  GULF LAND USA                 DALLAS           7,100.00
* TEXAS LIFE & CASUALTY CO.     DALLAS             543.21
  AFTON INDUSTRIES              DALLAS           1,234.51
  PALMOLIVE INNS                FORT WORTH         758.93
```

Literals

The report detail definition can also use a literal value. For example, underscores can print as the last characters of each line. You can specify the entry for this literal as shown in the following sample:

```
=>


-------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  -------------------------  - -   - - -  - - - -  - -- --- ------------
 ======  ========== T O P =======   = =   = = =  = = = =  = == === ============
 000300  BADDEBT                    _ _   _ _ _  _ _ _ _  N __ ___ ____________
 000400  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000500  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000600  CUSTOMER.OPEN$             _ _   _ _ _  _ _ _ _  _ __ ___ Z,ZZZ,ZZ9.99
 000700  '___'                      _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === ============
```

If the length of the literal exceeds the space provided on the screen, you can continue the literal on the next line by typing +0 in the TAB field of the continued line. For example, specify the literal "This is an example of continuation" as follows:

```
        Field Name, Literal,      Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
------  -------------------------  - -   - - -  - - - -  - -- --- ------------
000600  'This is an example of '   _ _   _ _ _  _ _ _ _  _ __ ___ ____________
000700  'continuation'             _ _   _ _ _  _ _ _ _  _ __ +0_ ____________
```

To ensure correct spacing when the two lines are concatenated, a space is included in the quotation marks at the end of the first line. If you leave out the space, you could specify the TAB as +1, rather than +0, to ensure correct spacing.

Functions or Mathematical Expressions

RDF lets you specify a field as a function or as a mathematical expression. For example, if the date is stored in the database as an alphanumeric value, the date must be formatted. You can include the value of ACTDT in the report using the $DATE function on the Detail fill-in:

```
=>


 ------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST           SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  -------------------------  - -   - - -  - - - -  - -- --- ------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === ============
 000400  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000500  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000600  CUSTOMER.OPEN$             _ _   _ _ _  _ _ _ _  _ __ ___ Z,ZZZ,ZZ9.99
 000700  $DATE('MM/DD/YY',;         _ _   _ _ _  _ _ _ _  _ 08 ___ ____________
 000800  DATE=ACTDT,TEM='YYMMDD')   _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === ============
```

Two lines are required to enter the function and its operands. A semicolon after the last character on the first line of the function entry indicates continuation. Also, a width is required for a date. A compiler error occurs if you do not specify a width.

The program code does not have to include the function in a procedure or define a variable to contain the result. The function is specified on the Detail fill-in and appears on the report output. You can define a name or label for the function. For example, you can assign FORMDATE as the label for the date function in the preceding example.

```
FORMDATE = $DATE('MM/DD/YY',DATE=ACTDT,TEM='YYMMDD')
```

You can then include the value n control break footings. See the section titled Headings and Footings later in this chapter for more information on using labeled expressions.

Note: The $OCC function is the only report function that you can specify as a field on the Detail fill-in. $OCC prints the count of primary records.

Including Fields from Multiple Dataviews

You can print data from more than one dataview on a single detail line by specifying the column name on the Detail fill-in. For example, a report named ORDRPT is created to print a list of all outstanding orders by order ID. The order identification number and the customer identification number are taken from the ORDER dataview. The customer name and salesman ID are taken from the CUSTOMER dataview. The field name qualifiers shown in the following screen indicate which dataview obtains the data.

```
=>
 -------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST         SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  -------------------------  - -   - - -  - - - -  - -- --- -------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === =============
 000400  ORDER.ORDID                _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 000500  ORDER.CUSTID               _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 000600  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 000700  CUSTOMER.SLMNID            _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

The column heading information is taken from the DataDictionary.

The following code generates the report. The PRODUCE statement is coded in the FOR FIRST CUSTOMER construct to ensure that the customer information prints with the order information.

```
FOR EACH ORDER
    FOR FIRST CUSTOMER
        WHERE CUSTOMER.CUSTID EQ ORDER.CUSTID
       PRODUCE ORDRPT
    ENDFOR
ENDFOR
```

The sequence in which the details print reflects the sequence in which the rows are retrieved from the database. The following sample shows the resulting detail lines:

```
ORDID CUSTID NAME                                 SLMNID

1021  A0130  SUN DIAL GROWERS                      29390
1023  A0150  IMPERIAL BANKCORP                     13150
1013  B0230  CHEMICAL MUTUAL                       00725
 .
 .
 .
```

Arranging the Report Data

You can specify the arrangement of report data across the page and down the page, using either the Parameter fill-in or the Detail fill-in.

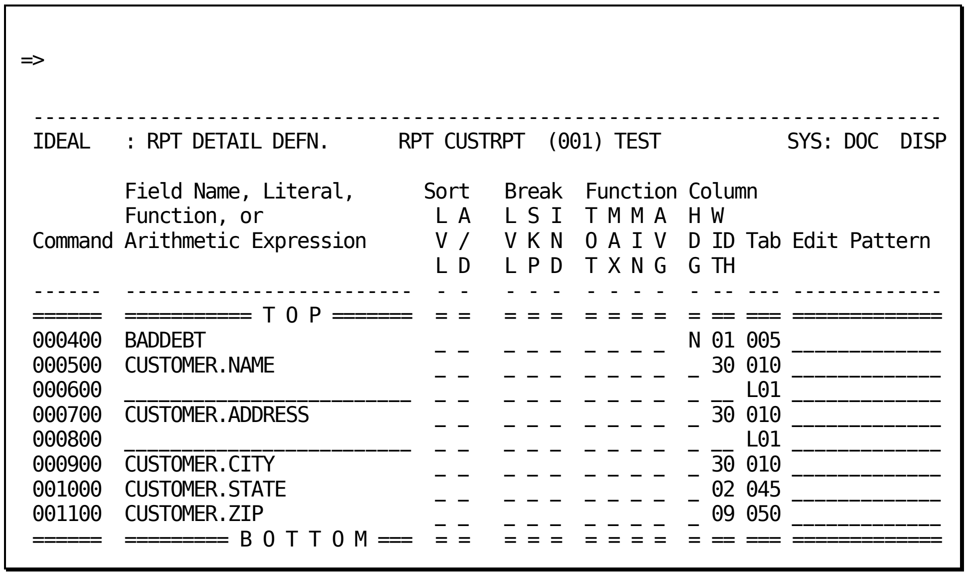
The Parameter fill-in defines the spacing between fields globally. You can specify a numeric value from 1 through 66. When you specify a numeric value, that number of spaces is inserted between the fields, regardless of the report width. You can specify automatic spacing using A. When you select automatic spacing, RDF calculates the space between each report column field in the following order:

1. Subtracting the total number of characters in the fields on the printed line from the report width.
2. Taking that result and dividing it equally between the fields.

The Detail fill-in specifies individual field placement. It overrides the Parameter fill-in specification for spaces between the fields.

Placing Individual Fields in the Detail Line

You can modify the default field placement by using the Detail fill-in screen. Look to the right of the following screen at the columns marked WIDTH and TAB. The values in these columns locate the fields on one or more print lines.



```
=>

-------------------------------------------------------------------------------
IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST           SYS: DOC   DISP

        Field Name, Literal,      Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
------  -------------------------  - -   - - -  - - - -  - -- --- -------------
======  =========== T O P =======  = =   = = =  = = = =  = == === =============
000400  BADDEBT                    _ _   _ _ _  _ _ _ _  N 01 005 _____________
000500  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ 30 010 _____________
000600  _________________________  _ _   _ _ _  _ _ _ _  _ __ L01 _____________
000700  CUSTOMER.ADDRESS           _ _   _ _ _  _ _ _ _  _ 30 010 _____________
000800  _________________________  _ _   _ _ _  _ _ _ _  _ __ L01 _____________
000900  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ 30 010 _____________
001000  CUSTOMER.STATE             _ _   _ _ _  _ _ _ _  _ 02 045 _____________
001100  CUSTOMER.ZIP               _ _   _ _ _  _ _ _ _  _ 09 050 _____________
======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

Defining Column Width

The width value specifies the maximum width of the field. If you do not specify a width value, RDF uses the largest of the following values:

- The actual width of the source field.
- The width of the column heading.
- The width of the column heading, plus 4 if summary information is requested for the field (for more information about summaries, see the section Generating the Bottom Line later in this chapter).
- The width specified by the edit pattern.

The width value should correspond to the edit pattern, but that is not required. Data is truncated if it cannot fit in the space allotted by the width. In the previous screen, the fields do not have a defined edit pattern.

The column width determines the length of the report field. It must be a positive integer. When using functions (such as $SUBSTR) to define a field, do not specify a width of zero.

Positioning with TAB Values

The TAB value can designate absolute or relative position, vertically or horizontally. In the previous example, all numeric values indicate an absolute tab position on the current print line. That is, BADDEBT prints in the fifth position, 05, and CUSTOMER.NAME prints in the tenth, 10, and so forth. You specify a relative position by preceding the numeric value with a plus sign. To print CUSTOMER.NAME five positions after the end of the previous field, the entry is +05. This provides the same result as the absolute positions shown on the previous sample screen. You can specify the tab value as MAX to print the field right-justified.

All tab values preceded with an L indicate vertical spacing, that is, advance print lines. You can specify the value as L followed by a numeric value from 0 through 99. L*nn* must be a separate tab entry. All other values on that line must be blank. L01 means advance to the next print line. L00 overtypes the current print line. This provides underscores and bold type. You can specify P to force a new page.

Result

The Detail fill-in definition repositions the fields on the output are shown in the following example. This multi-line sample is actually a single detail from the report.

```
CHEMICAL MUTUAL
555 MAIN STREET
FORT WORTH            TX    761026102
```

To obtain the output in this format, you do not have to modify the program code. A single PRODUCE statement generates one logical line of data. The detail definition simply reformats that data into multiple physical lines. This concept is very useful in generating mailing labels, checks, or invoices. (Appendix B contains a sample program to generate mailing labels.) When formatting the output, it might also be helpful to suppress automatic page breaks. This is done on the Parameter fill-in by specifying 0 as the Lines-per-page value.

Multiple lines produced by one detail line can have headings and aligned columns. Since this chapter focuses on how to position the fields, these examples suppress column headings by specifying N on the Parameter fill-in field Column Headings Desired. For information on controlling headings, see Headings and Footings later in this chapter.

Printing Multiple Lines with One Detail

You can define the Detail fill-in to produce multiple detail lines. Assume a report is generated that lists all orders for each customer. On the Detail fill-in, the information for customers prints on one line and the order number and date print on the next line. These two lines comprise one detail. A blank line is inserted between each line. The Detail fill-in is specified as shown in the following example:

```
=>
------------------------------------------------------------------------------
IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST         SYS: DOC  DISP

        Field Name, Literal,      Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
------  -------------------------  - -   - - -  - - - -  - -- --- -------------
======  =========== T O P =======  = =   = = =  = = = =  = == === =============
000400  CUSTOMER.CUSTID            _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000500  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000600  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000700  CUSTOMER.STATE             _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000800  CUSTOMER.ZIP               _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000900                             _ _   _ _ _  _ _ _ _  _ __ L01 _____________
001000  ORDER.ORDID                _ _   _ _ _  _ _ _ _  _ __ 008 _____________
001100  $DATE('MM/DD/YY',;         _ _   _ _ _  _ _ _ _  _ 08 +04 _____________
001200  DATE=ORDDT,TEM='YYMMDD')   _ _   _ _ _  _ _ _ _  _ __ ___ _____________
001300                             _ _   _ _ _  _ _ _ _  _ __ L01 _____________
======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

Notice the TAB value of 008 for the ORDER.ORDID field. This specification places the field at column 8 of the report, which moves the field to the next physical line. (Assume that one space is inserted between fields, and headings do not print.)

The following PDL code generates this report:

```
FOR EACH CUSTOMER
    FOR EACH ORDER
        WHERE ORDER.CUSTID EQ CUSTOMER.CUSTID
       PRODUCE CUSTRPT
    ENDFOR
ENDFOR
```

The following report shows the resulting report, with customer information printed for each order. Notice that there are two entries for customer A0130.

```
A0130 SUN DIAL CITRUS GROWERS    LOS ANGELES    CA 902130050
      1021    11/08/93

A0130 SUN DIAL CITRUS GROWERS    LOS ANGELES    CA 902130050
      1024    01/04/94

A0150 IMPERIAL BANKCORP          NEW YORK       NY 100190000
      1023    12/24/93

B0230 CHEMICAL MUTUAL            FORT WORTH     TX 761026102
      1013    11/05/94
.
.
.
```

Grouping Details for Individual Printing

All details are report groups, that is, a uniquely identified set of lines on the Detail fill-in. Each execution of the PRODUCE statement generates the entire group.

You can identify a report group with a group name on the Detail fill-in. For example, to name a group CUST, specify:

```
<<CUST>> GROUP
```

RDF allows multiple subgroups in the report group. It distinguishes these subgroups as primary and secondary. There can be only one primary group, but you can specify one or more secondary groups. When used with the PRODUCE statement, you can generate groups individually.

Primary and Secondary Groups

The first group specified on the Detail fill-in is the primary group, no matter whether it is named. In all of the examples thus far, only a primary group was defined. Secondary groups require a group name to distinguish them from the primary group and from each other. The keyword GROUP is optional.

For example, assume a report is generated listing all customers and their orders. The customer information is specified in the primary group named <<CUSTINF>>. The order information is specified in a secondary group named <<ORDINF>>. The following screen shows the definition of these two groups.

```
 =>
------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  -------------------------  - -   - - -  - - - -  - -- --- -------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === =============
 000400  <<CUSTINF>> GROUP          _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 000500  CUSTOMER.CUSTID            _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 000600  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 000700  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 000800  CUSTOMER.STATE             _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 000900  CUSTOMER.ZIP               _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 001000  _________________________  _ _   _ _ _  _ _ _ _  _ __ L01 _____________
 001100  <<ORDINF>> GROUP           _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 001000  ORDER.ORDID                _ _   _ _ _  _ _ _ _  _ __ 008 _____________
 001100  $DATE('MM/DD/YY',;         _ _   _ _ _  _ _ _ _  _ 08 +04 _____________
 001200  DATE=ORDDT,TEM='YYMMDD')   _ _   _ _ _  _ _ _ _  _ __ ___ _____________
 001300  _________________________  _ _   _ _ _  _ _ _ _  _ __ L01 _____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

When primary and secondary groups are defined, the program code must PRODUCE the groups individually. A group name is specified on the PRODUCE statement as report.group. If the group name is omitted, the primary group is produced.

```
FOR EACH CUSTOMER
    PRODUCE CUSTRPT.CUSTINF
     FOR EACH ORDER
        WHERE ORDER.CUSTID EQ CUSTOMER.CUSTID
       PRODUCE CUSTRPT.ORDINF
    ENDFOR
ENDFOR
```

The resulting output provides none, one, or several order details, but the customer information always prints. As you can see, customer number A0090 does not have any orders, while customer number A0130 has two orders. Multiple orders for a single customer print with the single line of customer information.

```
A0090 INTERNATIONAL BANK CORP    NEW YORK       NY 100059989

A0130 SUN DIAL CITRUS GROWERS    LOS ANGELES    CA 902130052
      1021    11/08/93
      1024    01/04/94

A0150 IMPERIAL BANKCORP          NEW YORK       NY 100190000
      1023    12/24/93

B0230 CHEMICAL MUTUAL            FORT WORTH     TX 761026102
      1013    11/05/94
 .
 .
 .
```

Primary Detail Group

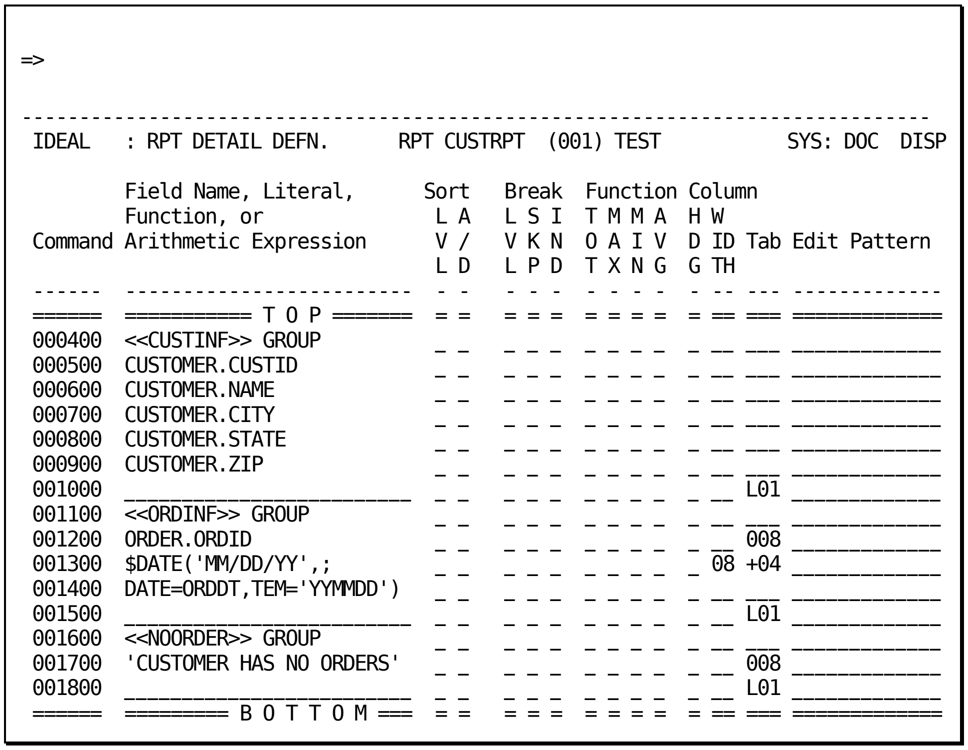
The generation of the primary detail group controls the contents of all other fields in the report. Each primary detail group produced is the owner of a larger set of lines, consisting of the following:

- Report heading
- Page heading
- Control break headings and footings
- Page footing
- Report footing
- One or more secondary detail groups, if any are produced for that primary group

The primary detail group should always be the first detail group produced for a report. When the primary detail group is produced, all of the above lines are produced with it. When the report prints, only the lines required, based on the position of the primary detail group in the report and on the page, print.

Multiple Secondary Groups

A primary group can have more than one secondary group. For example, the following Detail fill-in contains two secondary groups. The first defines the order identification number and date and the second defines a literal. The literal prints when a customer does not have any outstanding orders.



```
=>

------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  ------------------------   - -   - - -  - - - -  - -- --- ------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === ============
 000400  <<CUSTINF>> GROUP          _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000500  CUSTOMER.CUSTID            _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000600  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000700  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000800  CUSTOMER.STATE             _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000900  CUSTOMER.ZIP               _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 001000  _________________________  _ _   _ _ _  _ _ _ _  _ __ L01 ____________
 001100  <<ORDINF>> GROUP           _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 001200  ORDER.ORDID                _ _   _ _ _  _ _ _ _  _ __ 008 ____________
 001300  $DATE('MM/DD/YY',;         _ _   _ _ _  _ _ _ _  _ 08 +04 ____________
 001400  DATE=ORDDT,TEM='YYMMDD')   _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 001500  _________________________  _ _   _ _ _  _ _ _ _  _ __ L01 ____________
 001600  <<NOORDER>> GROUP          _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 001700  'CUSTOMER HAS NO ORDERS'   _ _   _ _ _  _ _ _ _  _ __ 008 ____________
 001800  _________________________  _ _   _ _ _  _ _ _ _  _ __ L01 ____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === ============
```

The following program prints the appropriate secondary group:

```
FOR EACH CUSTOMER
    PRODUCE CUSTRPT.CUSTINF
    FOR EACH ORDER
       WHERE ORDER.CUSTID EQ CUSTOMER.CUSTID
        PRODUCE CUSTRPT.ORDINF
    WHEN NONE
       PRODUCE CUSTRPT.NOORDER
    ENDFOR
ENDFOR
```

The output now appears as follows:

```
A0090 INTERNATIONAL BANK CORP    NEW YORK       NY 100059989
      CUSTOMER HAS NO ORDERS

A0130 SUN DIAL CITRUS GROWERS    LOS ANGELES    CA 902130052
      1021    11/08/93
      1024    01/04/94

A0150 IMPERIAL BANKCORP          NEW YORK       NY 100190000
      1023    12/24/93

B0230 CHEMICAL MUTUAL            FORT WORTH     TX 761026102
      1013    11/05/94
     .
     .
     .
```

Note: Produce the primary detail before any secondary details, otherwise, the output is unpredictable.

Sequencing the Report Data

In an online program that does not print a large number of rows, it is reasonable to allow the database to access the rows based on the WHERE clause criteria and the ORDERED BY clause.

```
FOR EACH CUSTOMER
    WHERE STATE EQ 'TX' AND CITY EQ 'DALLAS'
    ORDERED BY CUSTID
    PRODUCE CUSTRPT
ENDFOR
```

However, when you are going to print a large amount of data, you should run the program in batch. When the program executes in batch, the database can still control the sequence in which the rows are retrieved. In addition, RDF can sort the detail lines. RDF uses the system sort utility. You can specify any combination of up to nine fields. This is especially useful if you must sort on non-key fields.

In a CICS environment, CA CICSORT lets CA Ideal programs producing sorted reports run online.

Sorting Requirements

Sorting requirements are specified on the Detail fill-in. You can sort fields in ascending or descending sequence.

In the following example, sorting is requested on the value in STATE in ascending sequence. Notice the 1 typed in the LVL field under the SORT heading. The A in the field marked A/D next to the LVL field specifies an ascending sort.

```
=>
=>
=>
-------------------------------------------------------------------------------
IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST         SYS: DOC   DISP
        Field Name, Literal       Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
------  -------------------------  - -   - - -  - - - -  - -- --- -------------
======  ========== TOP ==========  = =   = = =  = = = =  = == === =============
000400  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000500  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000600  CUSTOMER.STATE             1 A   _ _ _  _ _ _ _  _ __ ___ _____________
000700  CUSTOMER.OPEN$             _ _   _ _ _  _ _ _ _  _ __ ___ Z,ZZZ,ZZ9.99_
======  ======== BOTTOM =========  = =   = = =  = = = =  = == === =============
```

The following report sample shows the output from Tennessee and Texas. Although all of the customers in Texas are listed together, they are not sorted any further:

```
SUNSTRAND BANKS             MEMPHIS     TN      932.00
BAY-BANK AUTOMOBILES        NASHVILLE   TN      543.21
CHEMICAL MUTUAL             FORT WORTH  TX      931.72
GULF LAND USA               DALLAS      TX    7,100.00
TEXAS LIFE & CASUALTY CO    DALLAS      TX      543.21
AFTON INDUSTRIES            DALLAS      TX    1,234.51
PALMOLIVE INNS              FORT WORTH  TX      758.93
SOUTHLAND STORES            HOUSTON     TX      342.91
UNION TRANSPORTATION        GALVESTON   TX        0.00
```

You can specify additional levels of the sort. In the following example, the primary group details are sorted by STATE, CITY, and then by NAME, all in ascending sequence.

```
=>
=>
=>
-------------------------------------------------------------------------------
IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST         SYS: DOC  DISP
        Field Name, Literal       Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
------  -------------------------  - -   - - -  - - - -  - -- --- -------------
======  ========== TOP ==========  = =   = = =  = = = =  = == === =============
000400  CUSTOMER.NAME              3 A   _ _ _  _ _ _ _  _ __ ___ _____________
000500  CUSTOMER.CITY              2 A   _ _ _  _ _ _ _  _ __ ___ _____________
000600  CUSTOMER.STATE             1 A   _ _ _  _ _ _ _  _ __ ___ _____________
000700  CUSTOMER.OPEN$             _ _   _ _ _  _ _ _ _  _ __ ___ Z,ZZZ,ZZ9.99_
======  ======== BOTTOM ========   = =   = = =  = = = =  = == === =============
```

The sort level indication determines the priority of each sort field. The sequence in which you specify the fields as details has no relevance. Now the report output for Texas prints as:

```
AFTON INDUSTRIES            DALLAS      TX    1,234.51
GULF LAND USA               DALLAS      TX    7,100.00
TEXAS LIFE & CASUALTY CO    DALLAS      TX      543.21
CHEMICAL MUTUAL             FORT WORTH  TX      931.72
PALMOLIVE INNS              FORT WORTH  TX      758.93
SOUTHLAND STORES            HOUSTON     TX      342.91
UNION TRANSPORTATION        GALVESTON   TX        0.00
```

Regardless of sort specifications, the primary group and its related secondary groups are maintained together. Each PRODUCE of a secondary group associates that detail with the most recent primary group detail.

You cannot compile reports that use RDF sort facilities online, but you must always run them in batch.

You can specify the sort as A for ascending or D for descending. Another collating sequence, such as that specified by CA-Sort or any of its clones, can be invoked. (Alternate sort packages must be defined at installation.) An alternate collating sequence (ACS) is invoked by specifying + for ascending and - for descending in the A/D field under the SORT heading of the Detail fill-in.

Formatting Based on Column Value Control Break

You can make a report more readable by including a physical break, such as advancing one or more lines or providing some notation between each new value in the break field. You can specify control breaks on primary detail lines to group the data based on the value in a specific column.

For example, a report can include a control break each time the value of STATE changes. In the following Detail fill-in, look at the column headed BREAK. Three fields are available to specify control break information.

- LVL or level specifies the control fields and their relative sequence. The field marked 1 is the highest level, 2 next, 3 next, and so on. A maximum of 9 levels is possible. In this example, there is one level, STATE.
- SKP designates the spacing between the control footing of one set and the control heading of the next. This value can be 1 through 9 lines inclusive, P to force a new page, or R to force a new page and begin page numbering from 1. In this example, SKP is specified as 2 to advance two lines.
- IND specifies whether the designated control field prints as part of the detail line. This does not affect whether the value prints in the control break footing. You can enter one of the following values:

N

Specifies the control field is not printed with detail line.

R

Specifies the control field is printed with every detail line.

Blank or G

Specifies the control field printed when value changes.

In this example, G specifies that the control field prints when the value of the field changes.

```
=>
--------------------------------------------------------------------------------
IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST       SYS: DOC  DISPLAY

        Field Name, Literal,      Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
------  -------------------------  - -   - - -  - - - -  - -- --- -------------
======  =========== T O P =======  = =   = = =  = = = =  = == === =============
000400  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000500  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000600  CUSTOMER.STATE             _ _   1 2 G  _ _ _ _  _ __ ___ _____________
000700  CUSTOMER.OPEN$             _ _   _ _ _  _ _ _ _  _ __ ___ Z,ZZZ,ZZ9.99_
======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

The following report segment only shows the output for the details under Tennessee, followed by a few details under Texas. This shows the control break as it appears between each state, the designated control break field. The value of the control break field is only listed with the first detail:

```
 .
 .
 .
SUNSTRAND BANKS            MEMPHIS     TN    932.00
BAY-BANK AUTOMOBILES       NASHVILLE         543.21
AFTON INDUSTRIES           DALLAS      TX  1,234.51
GULF LAND USA              DALLAS          7,100.00
TEXAS LIFE & CASUALTY CO   DALLAS            543.21
CHEMICAL MUTUAL            FORT WORTH        931.72
PALMOLIVE INNS             FORT WORTH        758.93
 .
 .
 .
```

The previous sample shows the output when you specify IND as G or is blank. The following output shows the difference when you specify IND as N. The data in the control break field is not included on the detail line:

```
 .
 .
 .
SUNSTRAND BANKS            MEMPHIS           932.00
BAY-BANK AUTOMOBILES       NASHVILLE         543.21
AFTON INDUSTRIES           DALLAS          1,234.51
GULF LAND USA              DALLAS          7,100.00
TEXAS LIFE & CASUALTY CO   DALLAS            543.21
CHEMICAL MUTUAL            FORT WORTH        931.72
PALMOLIVE INNS             FORT WORTH        758.93
 .
 .
 .
```

When you specify IND as R, the data in the control break field always prints on the detail line:

```
 .
 .
 .
SUNSTRAND BANKS            MEMPHIS     TN    932.00
BAY-BANK AUTOMOBILES       NASHVILLE   TN    543.21
AFTON INDUSTRIES           DALLAS      TX  1,234.51
GULF LAND USA              DALLAS      TX  7,100.00
TEXAS LIFE & CASUALTY CO   DALLAS      TX    543.21
CHEMICAL MUTUAL            FORT WORTH  TX    931.72
PALMOLIVE INNS             FORT WORTH  TX    758.93
 .
 .
 .
```

Identifying Control Breaks on the Report

The specific fields for control breaks are defined on the Detail Definition fill-in. The Parameter fill-in defines whether any notation, such as headings and footings, print to highlight the breaks. The prompts on the Parameter fill-in are shown in the following screen:

```
 =>
--------------------------------------------------------------- Partially shown
 IDEAL   : RPT PARAMETERS       RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

 .
 .
 .
Control break heading               N    (Y=Yes,N=No)
Control break footing               Y    (Y=Yes,N=No)
Automatic page footing summaries    N    (Y=Yes,N=No)
Group continuation at top of page   Y    (Y=Yes,N=No)
Annotated count in break footings   Y    (Y=Yes,N=No)
.
 .
 .
```

Parameters are described in the following sections.

Control Break Headings

The control break heading prints on a line by itself immediately preceding the detail lines. The heading is composed of asterisks followed by the specified or default column heading for the control break field and the control value. To specify a column heading, use the Column Headings fill-in. See Defining Column Headings later in this chapter.

For example, assume CUSTOMER.STATE is defined as the control break field and control break headings are requested. Each time the value of CUSTOMER.STATE changes, a control break heading prints. For the state of Texas, the heading prints the following:

```
** STATE TX
```

Headings also print for any additional control break levels. So if you specify CUSTOMER.CITY as a second level control break, a heading prints each time the value of CUSTOMER.CITY changes. Since two asterisks are added for each control break level, it prints the following:

```
**** CITY DALLAS
```

You can define the text for the control break heading on the Heading fill-in to override the default heading text. See Headings and Footings section.

Control Break Footings

A control break footing consists of the specified or default column heading for the control break field and the value of the field. For example, assume CUSTOMER.STATE is defined as the control break field and control break footings are printed. When the value of CUSTOMER.STATE is TX for Texas, the footing prints the following:

```
STATE TX
```

You can define the text for the control break footing on the Heading fill-in to override the default footing text. See Headings and Footings later in this chapter.

Control Break Continuation

When the details for an individual control group overflow the page, it is helpful to provide an identifying notation at the top of the new page. Specify on the Parameter fill-in whether a group continuation heading should print at the top of a new page. If you specify yes, the heading prints regardless of whether a control break occurred.

To distinguish a continuation from an actual control break, (CONT.) is included in the heading:

```
** STATE TX (CONT.)
```

If a continuation is not requested, the control break heading prints only when a control break occurs.

Annotated Control Break Footings

It is often helpful to document how many group details were printed in a control group. RDF automatically accumulates and prints that value as part of the footing if requested to do so on the Parameter fill-in.

For example, if you request annotation on the previous control footing example and fifteen customer detail lines were printed for Texas, the footing prints as the following:

```
STATE TX
COUNT               15
```

You can obtain additional types of information based on the control group. This summary information is described in the Generating the Bottom Line section. Summary information, including COUNT, is most meaningful when the records are sorted according to the control break field. This means that the FOR construct must include an ORDERED BY clause to access the records in the required sequence.

Note: To combine these examples, assume the report contains the customer name, city, state, and outstanding amount owed. A control break is specified on CUSTOMER.STATE. A second level control break is specified on CUSTOMER.CITY. The FOR construct accesses the data ordered by state and then by city. Control break headings and footings are to be printed. Control break footings are to contain annotated totals. The value of CUSTOMER.STATE is not to print on the detail line as specified on the Detail fill-in. The value of CUSTOMER.CITY prints only when the value changes. The Detail fill-in is defined as shown in the following screen:

```
=>
------------------------------------------------------------------------------
IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

        Field Name, Literal,      Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
------  ------------------------   - -   - - -  - - - -  - -- --- -------------
======  =========== T O P =======  = =   = = =  = = = =  = == === =============
000300  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  _ __ ___ _____________
000400  CUSTOMER.CITY              _ _   2 2 G  _ _ _ _  _ __ ___ _____________
000500  CUSTOMER.STATE             _ _   1 2 N  _ _ _ _  _ __ ___ _____________
000600  CUSTOMER.OPEN$             _ _   _ _ _  _ _ _ _  _ __ ___ _____________
======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

The output for Texas prints as the following:

```
.
.
.
      ** STATE TX
      **** CITY DALLAS
      GULF LAND USA             DALLAS           7,100.00
      TEXAS LIFE & CASUALTY                        543.21
      AFTON INDUSTRIES                           1,234.00
           CITY DALLAS
           COUNT         3

      **** CITY FORT WORTH
      CHEMICAL MUTUAL           FORT WORTH         931.72
      PALMOLIVE INNS                               758.93
           CITY FORT WORTH
           COUNT          2
           STATE TX
           COUNT          5
.
.
.
```

Even though column headings were suppressed on the Parameter fill-in, the control break headings and footings print for each level of control break, using the default column headings.

Headings and Footings

The usefulness of the data in reports is enhanced by descriptive text provided in headings and footings. Reports generated by RDF can include a variety of headings and footings to identify the entire report or specific data.

- You can define simple page headings on the Parameter fill-in to include at the top of each page.
- You can define more complex page headings, page footings, report headings, and report footings on the Heading fill-in screen.
- RDF automatically generates simple column headings for the fields specified on the Detail fill-in.
- You can define complex column headings on the Column Heading fill-in.

Defining Report and Page Headings and Footings

Report and page headings and footings provide general information about the report.

- Report heading prints only on the first page of the report.
- Page heading prints at the top of every page.
- Report footing prints as the last entry on the last page.
- Page footing prints at the bottom of every page.

Producing Headings and Footings

Headings and footings are generated only when a primary detail group is produced. The report and page headings are generated when the primary detail group is produced. The report and page footings are updated when the current primary detail group is completed, which is determined by the following:

- Beginning of the next primary detail group
- Release of the report
- End of the program

Until the primary detail group is completed, the report and page footings contain values from the previous primary detail group.

Although each primary detail group has a report heading, page heading, page footing, and report footing associated with it, only selected headings and footings actually print.

- The report heading associated with the first primary detail group in the report prints.
- The report footing associated with the last primary detail group in the report prints.
- The page headings associated with the first primary detail group on each page prints.
- The page footings associated with the last completed primary detail group that fits on each page prints.

Simple Page Headings

The Parameter fill-in defines a simple page heading. A simple page heading is defined as one that:

- Is 42 characters or less in length.
- Occupies a single line.
- Is positioned on the right, left, or center of the line.
- Is a literal string.

The Parameter fill-in prompts for the literal and the position where it prints. It is the last item on the Parameter fill-in.

Assume that a report containing a list of customers is to print. You can define a simple page heading as the following:

```
=>
 ------------------          ------------ Partially shown
 IDEAL   : RPT PARAMETERS       RPT CUSTRPT  (001) TEST         SYS: DOC  DISP

 .
 .
 .
 Page Heading
   Heading            Customer List_____________________________
   Position           C    (C=Center, L=Left Justify, R=Right Justify)
```

The position was specified as C, meaning RDF centers the text on the line at the top of every page.

Date and Page Numbers

Although the current date and page numbers are not actually considered a part of the simple page heading, take care when specifying their placement on the page. The page heading, current date, and page numbers should not overlay each other.

If the date and page numbers are to print at the top of the page along with the heading, the specification on the Parameter fill-in is shown in the following screen:

```
=>
 --------------------------------------------------------------- Partially shown
 IDEAL   : RPT PARAMETERS       RPT CUSTRPT  (001) TEST          SYS: DOC  DISP
 .
 .
 .
 Date
   Position          TL   (NO=None, BR=Bot.Right, BL=Bot.Left, BC=Bot.Ctr.,
                           TR=Top Right, TL=Top Left, TC=Top Center)
   Format            MM/DD/YY

 Page Numbers
   Position          TR
   Format            H    (D=Digits Only, H=With Hyphens, P= Page      nnn)

 Page Heading
   Heading           Customer List_____________________________
   Position          C    (C=Center, L=Left Justify, R=Right Justify)
```

This appears at the top of page 1 as the following:

```
02/28/95                       Customer List                                -1-
```

You must specify the date and page number format on the Parameter fill-in. You can specify any valid date format. You can print the page number as digits only, as digits surrounded with hyphens (as shown in the preceding example), or as the word Page followed by digits.

A variety of placement specifications are available, including placing the date and page number at the top and bottom of the page. The possible placements are listed on the Parameter fill-in screen next to the Position prompt. For information on how to specify a position other than those available on the Parameter fill-in, see User-defined Date and Page Number Location later in this chapter.

Complex Page Headings

You can define complex page headings on the Report Heading fill-in. This screen lets you closely control the text and format of the heading. RDF handles the entire page heading as a single entity. Therefore, you can define multi-line headings on this screen.

As with defining details for the report, a field is provided for the heading text. You can specify it as a literal, a variable, a function, or an arithmetic expression.

The following definition provides an example of a multi-line page heading composed of a function, two literals, a line break, and a working data variable. The components of the heading are entered vertically on the fill-in, but appear horizontally across the top of each page of the report. Specifications in the TAB field on the screen control the actual spacing on the printed page, horizontal and vertical.

```
 =>
-------------------------------------------------------------------------------
 IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST           SYS: DOC  DISP

          Field Name, Literal, Function,                    Column
 Command  or Arithmetic Expression                          Wid Tab Edit Pattern
 ------   ------------------------------------------------  --  --- -------------
 ======   ==================== T O P ===================== ==  === =============
 000300   $DATE('MM/DD/YY')                                 08  001 _____________
 000400   'Customer List'                                   __  +02 _____________
 000500   ________________________________________________  __  L01 _____________
 000600   'Reporting States:'                               __  011 _____________
 000700   STNAMES                                           25  +02 _____________
 ======   ===================== B O T T O M ============== ==  === =============
```

Notice the use of the TAB field to skip lines (L01), to position the data in absolute terms (11), and in relative terms (02). The multi-line heading defined in this example appears as the following:

```
02/28/95  Customer List
          Reporting States:  CA, IL, MD, NY, OH and TX
```

Notice that the working data variable STNAMES, specified in the definition above, provides a list of the states included in the report. Make sure that the placement of the page heading does not conflict with the placement of page numbers or page dates.

User-Defined Date and Page Number Location

You can control the location of the date and page number on the Heading fill-in, overriding any placement specified on the Parameter fill-in. CA Ideal provides functions to specify the date or page number in the page heading or footing definition. The Parameter fill-in controls the edit pattern for these functions. The functions are:

- $RPT-DATE-Date of the report. The date value is obtained from an ASSIGN REPORT statement, if one is specified. Otherwise, the current date is used.
- $RPT-PAGE-Page number. The page value is obtained from an ASSIGN REPORT statement, if one is specified. Otherwise, a value of 1 is used.

For example, to print the report date and page number right justified on the top of each page as the following:

```
                                                                   Customer List
                                                                      02/28/2006
                                                                             -1-
```

The Heading fill-in contains an entry for the page heading as follows:

```
 =>
-------------------------------------------------------------------------------
 IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal, Function,                  Column
 Command or Arithmetic Expression                        Wid Tab Edit Pattern
 ------  ----------------------------------------------- --  --- -------------
 ======  ==================== T O P ===================== ==  === =============
 000300  <<PH>>                                          __  ___ _____________
 000400  'Customer List'                                 __  MAX _____________
 000500  _______________________________________________ __  L01 _____________
 000600  $RPT-DATE                                       __  MAX _____________
 000700  _______________________________________________ __  L01 _____________
 000800  $RPT-PAGE                                       __  MAX _____________
 ======  ===================== B O T T O M ============= ==  === =============
```

TAB values print the heading on multiple lines. MAX right-justifies the data.

Other Report Heading Fillin Facilities

The Report Heading fill-in can also define the page footing, report heading, report footing, and control break headings and footings. By default, RDF assumes that the first definition on the Report Heading fill-in is a page heading.

Labels distinguish each type of heading and footing. You must specify the label on a line by itself. The labels are specified with the following:

- **<<PH>>-**Page heading to print at the top of every page. Specify the label if page heading is not the first or only definition on the Report Heading fill-in.
- **<<PF>>-**Page footing to print on the bottom of every page.
- **<<RH>>-**Report heading to print on a separate page at the beginning of the report.
- **<<RF>>-**Report footing to print on the last page of the report after the report summary information. It can be designated to print on a separate page.
- **<<BH>>-**Controls break heading to print as a heading before the first line of each control break group. The heading is defined for a specific control break level.
- **<<BF>>-**Controls break footing to print after the last line of each control break group. The footing is defined for a specific control break level.

The following text describes user-defined report and page headings and footings. A description of user-defined control break headings and footings follow. You can include several functions in the page footing and the control break footing. Since these functions provide summary values based on page breaks or on control breaks, they are described in Generating the Bottom Line later in this chapter.

Defining Report and Page Headings and a Report Footing

You can define multiple headings and footings at one time on the Report Heading fill-in. For example, the following Report Heading fill-in specifies that a report heading prints on a separate page before the report, a page heading prints on every page, and a report footing prints on a separate page after the report.

```
=>
-------------------------------------------------------------------------------
IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal, Function,                 Column
Command  or Arithmetic Expression                       Wid Tab Edit Pattern
------   ---------------------------------------------- --  --- ------------
======   ==================== T O P ================== ==  === ============
000300   <<RH>>                                         __  ___ ____________
000400   _______________________________________________ __  L10 ____________
000500   'Sample Report for Manual'                     __  043 ____________
000600   _______________________________________________ __  L06 ____________
000700   'Control Break is on State'                    __  043 ____________
000800   <<PH>>                                         __  ___ ____________
000900   $DATE('MM/DD/YY')                              08  001 ____________
001000   'Customer List'                                11  +02 ____________
001100   _______________________________________________ __  L01 ____________
001200   'Reporting States:'                            __  011 ____________
001300   STNAMES                                        25  +02 ____________
001400   <<RF>>                                         __  ___ ____________
001500   'Report Footing With Page Eject'               __  041 ____________
001600   _______________________________________________ __  L08 ____________
001700   'That's All Folks!'                            __  046 ____________
======   ===================== B O T T O M ============ ==  === ============
```

Define page headings and footings on the Report Heading fill-in or on the Parameter fill-in, not on both. When using the Report Heading fill-in, specify only one label of each type.

When accumulating totals for the report footing or the page footing, the values are calculated for every PRODUCE statement. The last report footing for the last row is written to the report. Thus, if the rows are sorted when accessed, that is by an ORDERED BY clause, the totals should print properly. The last produced row that contains the valid total value is the last row printed.

Note: The number of lines skipped after the page heading is specified as a global value on the Parameter fill-in. You can specify the value as 0 through 9 on the Parameter fill-in. Since this value is also used for column headings, there might be instances when it is necessary to override this value for page headings. This can be done using the TAB column on the Report Heading fill-in.

As with report groups on the Detail fill-in, each set of specifications for a heading or footing is treated as a single entity. Therefore, RDF handles the TAB specification as part of the page heading itself and then applies the global value, number of lines to skip, to the entire page heading.

For example, the value on the Parameter fill-in is 1, but 5 lines are skipped after a page heading. The entry in the TAB column is the total number of lines to skip, that is 5, less the automatic number of lines skipped, that is 1. So, L04 is entered in the TAB column for the <<PH>> heading.

```
 =>
-------------------------------------------------------------------------------
 IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST         SYS: DOC  DISP

         Field Name, Literal, Function,                 Column
 Command  or Arithmetic Expression                       Wid Tab Edit Pattern
 ------   ---------------------------------------------- --  --- -------------
 ======   =================== T O P ==================== ==  === =============
 000300   <<PH>>                                         __  ___ _____________
 000400   'Customer List'                                13  ___ _____________
 000500   ______________________________________________ __  L04 _____________

======= ==================== BOTTOM =========== ==  === =============
```

Defining Control Break Headings and Footings

You can override the default text printed for control break headings and footings on the Heading fill-in. The headings and footings are defined independently for each control break level by specifying the level along with the group label. The default headings and footings print for any control break levels that were not overridden on the Heading fill-in. Thus, automatic headings and footings for some levels can be interspersed with user-defined headings and footings for other levels.

For example, a level-1 control break is specified on the Detail fill-in for the value in CUSTOMER.STATE. Rather than print the value with two asterisks as in "** STATE TX", the control break heading prints as Current State: followed by the full state name. A working data variable STNAME contains the actual state name.

This is specified on the Heading fill-in as shown in the following screen:

```
 =>
-------------------------------------------------------------------------------
 IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal, Function,                Column
 Command or Arithmetic Expression                      Wid Tab Edit Pattern
 ------  ----------------------------------------------- --  --- -------------
 ======  ==================== T O P ===================== ==  === =============
 000300  <<BH>> LEVEL 1                                   __  ___ _____________
 000400  'Current State:'                                 __  ___ _____________
 000500  STNAME                                           __  +02 _____________
 ======  ===================== B O T T O M ============== ==  === =============
```

When the current level-1 control break is for Texas, the control break heading prints as the following:

```
Current State:  TEXAS
```

You can define a control break heading for any level by specifying the level with the heading. You can specify the contents of the heading with literal text, variables, or expressions. You cannot specify a subscripted variable. You can define a single heading for multiple control breaks by including the level number with the label. For example, the following defines a single control break heading to print at the break for each level-1, level-2, and level-3 control break.

```
<<BH>> LEVEL 1 2 3
```

Control break footings are defined similar to control break headings. The group label is followed by the break level. For example, the following defines a control break footing to print at the end of each level-2 control break.

```
<<BF>> LEVEL 2
```

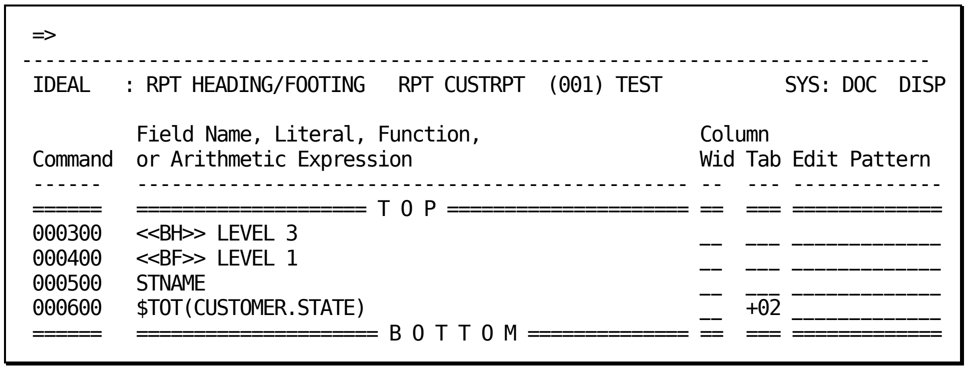
Use level-0 to obtain grand totals at the end of the report.

For information on using functions in the control break footing, see Generating the Bottom Line later in this chapter.

Suppressing Control Break Headings and Footings

You can suppress control break headings and footings for one or more control break levels by specifying the label for a user-defined control break heading or footing on the Heading fill-in with no lines following.

For example, to suppress a control break heading for a level-3 control break, the label is specified and a heading is not. Notice in the following screen that the control break heading label line is immediately followed by another label to define a control break footing for level-1.



```
=>
-------------------------------------------------------------------------------
IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST         SYS: DOC  DISP

         Field Name, Literal, Function,                   Column
Command  or Arithmetic Expression                         Wid Tab Edit Pattern
------   ------------------------------------------------ --  --- -------------
======   ==================== T O P ==================== ==  === =============
000300   <<BH>> LEVEL 3                                   __  ___ _____________
000400   <<BF>> LEVEL 1                                   __  ___ _____________
000500   STNAME                                           __  ___ _____________
000600   $TOT(CUSTOMER.STATE)                             __  +02 _____________
======   ===================== B O T T O M ============== ==  === =============
```

Note: If you define your own control break headings or footings, you must account for every break level by defining or suppressing control break headings and footings. Also, if you define your own control break heading or footing for a report that contains only one break, you must specify the level 1 clause.

You can turn off the control break headings and footings in the Parameters fill-in, but if you specify them in the Heading fill-in, be sure to specify yes on the corresponding Parameters fill-in. Otherwise, the report output might be incorrect.

Defining Column Headings

By default, RDF automatically includes headings for every defined field. For CA Datacom/DB Database, the headings for dataview columns are derived from the specifications stored for the column in the database. The default column headings contain a maximum of 20 characters since the cataloged dataview truncates the 36-character dataview heading to 20 characters. If a heading was not defined, RDF uses the column name.

Specifications on the Parameter fill-in define whether column headings print on the report and how those headings are formatted. The prompts are shown in the following screen:

```
=>
 --------------------------------------------------------------- Partially shown
 IDEAL   : RPT PARAMETERS       RPT CUSTRPT  (001) TEST           SYS: DOC  DISP

   .
   .
 Spacing after page and column hdgs   1    (0 thru 9)
   .
   .
 Column headings desired              Y    (Y=Yes,N=No)
 Column headings indication           U    (U=Underscore,N=None,D=Dashes)
   .
```

You can specify whether column headings print. For example, column headings do not print in the sample program to print mailing labels as shown in the section in the appendix "Sample Reports." Another sample program defines multiple detail lines and prints column headings.

You can print the column heading with underscores or dashes or without a special indicator:

```
Underscores            Dashes       None
___________            ------
```

The number of lines automatically skipped after the column headings is defined on the Parameter fill-in. It is the same value used for page headings. This value can be from 0-9, inclusive.

User-Defined Column Headings

The Parameter fill-in defines global placement attributes for column headings. You can control the content and location of the column headings using the Column Heading fill-in. First you must specify on the Parameter fill-in that column headings are to print. Then, on the Detail fill-in, type a U in the field marked HDG to designate individual fields whose headings are to user-defined, as shown in the following screen:

```
 =>
------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  -------------------------  - -   - - -  - - - -  - -- --- ------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === ============
 000300  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  U __ ___ ____________
 000400  CUSTOMER.ADDRESS           _ _   _ _ _  _ _ _ _  U __ ___ ____________
 000500  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000600  CUSTOMER.STATE             _ _   1 2 _  _ _ _ _  _ __ ___ ____________
 000700  CUSTOMER.ZIP               _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000800  CUSTOMER.OPEN$             _ _   _ _ _  A _ _ _  U __ ___ Z,ZZZ,ZZ9.99_
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === ============
```

If you want a column heading to appear for arithmetic expressions and functions, you must specify a user-defined heading. RDF can obtain default headings for columns (from the database or column name) and for variables (from the name of the variable).

Once a U is specified on the Detail fill-in, the Column Headings fill-in screen is accessed. This screen provides an area to define the headings. Notice that, in the column headings for each field in the following screen, uppercase and lowercase alphabetic characters are used.

```
 =>
--------------------------------------------------------------------------------
 IDEAL   : RPT COLUMN HEADINGS   RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Column
         Function, or              H W
 Command Arithmetic Expression     D ID Tab Headings
                                   G TH
 ------  ------------------------- - -- --- -----------------------------------
 ======  =========== T O P ======= = == === ===================================
 000400  CUSTOMER.NAME             U __ ___ /     Name_________________________
 000500  CUSTOMER.ADDRESS          U __ ___ Full Address_______________________
 000600  CUSTOMER.CITY             _ __ ___ ___________________________________
 000700  CUSTOMER.STATE            _ __ ___ ___________________________________
 000800  CUSTOMER.ZIP              _ __ ___ ___________________________________
 000900  CUSTOMER.OPEN$            U __ ___ /Outstanding/Amount Owed___________
 ======  ========= B O T T O M === = == === ===================================
```

Column headings are defined only for those fields that have a U in the area marked HDG. The Column HDG, WIDTH, and TAB fields are carried over from the Detail fill-in. The default column headings print for the other columns.

Including Blanks in the Heading

You can include blanks in the headings. For example, the heading for the column named NAME is the word Name centered over the data. This is accomplished by specifying leading blanks in the heading. To include leading blanks as part of the literal, a special character, here the back slash, is used. The heading for the second column, ADDRESS, does not require a special leading character. The blanks are embedded to provide a two-word heading, "Full Address."

Defining Multiline Column Headings

You can define multi-line column headings by specifying a line break in the heading. In the previous example, the last heading prints on more than one line. This is specified by the slashes (/) that delimit the lines.

Actually, when the first character of the heading literal is not alphabetic or numeric, that character is assumed to be a delimiter. Each successive occurrence of that character forces the following text of the heading to a new line. Thus, the heading for $OPEN prints as the following:

```
Outstanding
Amount Owed
```

A heading can print on a maximum of five lines.

There can be instances when the first character in the heading literal is normally considered a delimiter character. To ensure that the special character is included in the text, specify another character as the delimiter. For example, if asterisks are included in the heading, use a slash as the delimiter.

```
/***** Amount Owed *****
```

Continuation

You can continue the text of a heading on three lines at most. This is done by typing a semicolon after the last character of the field name. For example, assume the following three-line heading is defined for the column OPEN$:

```
Outstanding
Amount Owed
By Customer
```

This is specified as the following:

```
000900  CUSTOMER.OPEN$;            U __ ___ /Outstanding/Amount Owed/___________
001000  _________________________ _ __ ___ By Customer_________________________
```

Notice the semicolon after the column name, CUSTOMER.OPEN$. The slashes used as delimiters cause the heading itself to print on multiple lines.

Suppressing Print of Specific Column Headings

You can suppress individual column headings, overriding the global specification on the Parameter fill-in to print the headings. This is done by specifying N in the HDG column on the Detail fill-in.

For example, on the following Detail fill-in screen, column headings do not print for the columns CITY, STATE, and ZIP:

```
 =>
-------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  -------------------------  - -   - - -  - - - -  - -- --- ------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === ============
 000300  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  U __ ___ ____________
 000500  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  N __ ___ ____________
 000600  CUSTOMER.STATE             _ _   1 2 _  _ _ _ _  N __ ___ ____________
 000700  CUSTOMER.ZIP               _ _   _ _ _  _ _ _ _  N __ ___ ____________
 000800  CUSTOMER.OPEN$             _ _   _ _ _  A _ _ _  U __ ___ Z,ZZZ,ZZ9.99
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === ============
```

Defining Column Headings for Secondary Groups

The Column Heading fill-in can only define headings for the columns in the primary group. You can specify column headings for secondary group fields as literals on the Detail fill-in.

For example, assume that a report is generated that prints information from every CUSTOMER row along with every ORDER for the customer. The primary group consists of the customer information. A secondary group consists of the order information. Headings for the secondary group are specified as part of the primary group. The secondary group headings or literals print when the customer information is output.

The Detail fill-in contains the specifications shown in the following example:

```
=>
 -------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  ------------------------  - -   - - -  - - - -  - -- --- ------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === ============
 000200  <<PRIMARY>> GROUP          _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000300  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  U __ ___ ____________
 000400  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000500  CUSTOMER.STATE             _ _   1 2 _  _ _ _ _  _ __ ___ ____________
 000600  CUSTOMER.OPEN$             _ _   _ _ _  _ _ _ _  U __ ___ Z,ZZZ,ZZ9.99
 000700  'Ord #'                    _ _   _ _ _  _ _ _ _  _ 05 012 ____________
 000800  'Ord Date'                 _ _   _ _ _  _ _ _ _  _ 08 019 ____________
 000900  '_____'                    _ _   _ _ _  _ _ _ _  _ 05 012 ____________
 001000  '________'                 _ _   _ _ _  _ _ _ _  _ 08 019 ____________
 001200  ________________________   _ _   _ _ _  _ _ _ _  _ __ L02 ____________
 001300  <<ORDER>>                  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 001400  ORDER.ORDID                _ _   _ _ _  _ _ _ _  _ 05 012 ____________
 001500  $DATE('MM/DD/YY',;         _ _   _ _ _  _ _ _ _  _ 08 019 ____________
 001600  DATE=ORDDT,TEM='YYMMDD')   _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === ============
```

Notice how the TAB values line up the headings Ord # and Ord Date with the ORDER.ORDID field and $DATE function in the secondary group. The underscores, which also line up with the headings, provide a clear separation between the headings and the data. The TAB value L02 as the last specification in the primary group forces a blank line between the heading and the data from the secondary group.

In the program, the PRODUCE statement prints the primary and secondary groups. The secondary group prints multiple times based on the number of outstanding orders. The following code shows the pertinent segment:

```
FOR EACH CUSTOMER
   PRODUCE CUSTRPT.PRIMARY
   FOR EACH ORDER
       PRODUCE CUSTRPT.ORDER
   ENDFOR
ENDFOR
```

This program sample assumes that every customer has at least one order. Since this might not be a valid assumption, you can specify the secondary group heading information as another secondary group. Then printing the headings for the order details is controlled based on whether there are any orders. To define headings as a secondary group, precede the heading detail with a group label, as shown in the following:

```
=>
 ------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  ------------------------   - -   - - -  - - - -  - -- --- ------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === ============
 000100  <<PRIMARY>> GROUP          _ _   _ _ _  _ _ _ _  U __ ___ ____________
 000200  CUSTOMER.CUSTID            _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000300  CUSTOMER.CUSTNAME          _ _   _ _ _  _ _ _ _  U __ ___ ____________
 000400  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000500  CUSTOMER.STATE             _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000600  CUSTOMER.OPEN$             _ _   _ _ _  _ _ _ _  U __ ___  Z,ZZZ,ZZ9.99
 000700  <<ORDHEAD>> GROUP          _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 000800  'Ord #'                    _ _   _ _ _  _ _ _ _  _ 05 012 ____________
 000900  'Ord Date'                 _ _   _ _ _  _ _ _ _  _ 08 019 ____________
 001000  '_____'                    _ _   _ _ _  _ _ _ _  _ 05 012 ____________
 001100  '________'                 _ _   _ _ _  _ _ _ _  _ 08 019 ____________
 001200                             _ _   _ _ _  _ _ _ _  _ __ L02 ____________
 001300  <<ORDER>> GROUP            _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 001400  ORDER.ORDID                _ _   _ _ _  _ _ _ _  _ 05 012 ____________
 001500  $DATE('MM/DD/YY',;         _ _   _ _ _  _ _ _ _  _ 08 019 ____________
 001600  DATE=ORDDT,TEM='YYMMDD')   _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === ============
```

The code is modified to include an IF construct to test a flag, FIRST-FLAG, defined in working data. Based on the value of the flag, the secondary group heading is output as needed:

```
FOR EACH CUSTOMER
   PRODUCE CUSTRPT.PRIMARY
   SET FIRST-FLAG = TRUE
   FOR EACH ORDER
       WHERE ORDER.CUSTID EQ CUSTOMER.CUSTID
       IF FIRST-FLAG
          PRODUCE CUSTRPT.ORDHEAD
          SET FIRST-FLAG = FALSE
       ENDIF
       PRODUCE CUSTRPT.ORDER
   ENDFOR
ENDFOR
```

The following screen is a sample of the output:

```
Id      Name                    City         State         Zip Code

A0120  International Bank Corp  New York     NY            100059989

A0130  Sun Dial Citrus Grove    Los Angeles  CA            902130052

       Ord#            Ord Date

       1021            11/08/93
       1024            01/04/94

A0150  Imperial Bankcorp        New York     NY            100190000

       Ord#            Ord Date

       1023            10/24/93

A0230  Chemical Mutual          Fort Worth   TX            761026102



Ord#            Ord Date

       1013            11/05/94
```

Generating the Bottom Line

There is a wide variety of evaluated and summary information that can be generated automatically using RDF without additional code and working data variables. This information includes totals, maximums, minimums, and averages for one or more fields in the report. Although these fields do not have to be control break fields, the values are obtained based on the control break group and the entire report.

The Detail fill-in defines which values are obtained for the fields. You can print the values as an annotation, each on a separate line, or as a single summary line.

Identifying the Function

The Detail fill-in specifies which function executes for the field.

For example, a report is to print all of the customers in the CUSTOMER table. The report includes the total amount of accounts receivable (total OPEN$) from all customers in each state. The value is annotated and displayed at the control break.

To accomplish this, you must specify the following:

- The control break on CUSTOMER.STATE.
- The request to accumulate the total accounts receivable under the TOT Function heading on the Detail fill-in for the column CUSTOMER.OPEN$. The A in the field specifies that the total displays as annotated.

The following screen shows the Detail fill-in for this report:

```
=>
-------------------------------------------------------------------------------
IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

        Field Name, Literal,      Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
------  -------------------------  - -   - - -  - - - -  - -- --- -------------
======  =========== T O P =======  = =   = = =  = = = =  = == === =============
000400  CUSTOMER.NAME          _ _   _ _ _  _ _ _ _  U __ ___ _____________
000500  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  U __ ___ _____________
000600  CUSTOMER.STATE             _ _   1 2 N  _ _ _ _  _ __ ___ _____________
000700  CUSTOMER.OPEN$             _ _   _ _ _  A _ _ _  U __ ___ Z,ZZZ,ZZ9.99_
======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

The column headings are user-defined. The following example shows the resulting details and annotated total for the states of Tennessee and Texas only.

```
Customer Name           City         Open Dollar

** STATE TN
SUNSTRAND BANKS         MEMPHIS           932.21
BAY-BANK AUTOMOBILES    NASHVILLE         123.15

     TOTAL Open Dollar        1,055.36

** STATE TX
AFTON INDUSTRIES        DALLAS          1,234.51
GULF LAND USA           DALLAS          7,100.00
TEXAS LIFE & CASUALTY   DALLAS            543.21
CHEMICAL MUTUAL         FORT WORTH        931.72
PALMOLIVE INNS          FORT WORTH        758.93
UNION TRANSPORTATION    GALVESTON           0.00
SOUTHLAND STORES        HOUSTON           342.91

     TOTAL Open Dollar       10,911.28
```

You can specify the value to print on a summary line beneath the corresponding column by typing S instead of A under the appropriate function name on the Detail Fill-in. Looking at just the states of Tennessee and Texas, this prints as the following:

```
Customer Name              City          Open Dollar

** STATE TN
SUNSTRAND BANKS            MEMPHIS            932.21
BAY-BANK AUTOMOBILES       NASHVILLE          123.15

TOTAL                                       1,055.36

** STATE TX
AFTON INDUSTRIES           DALLAS           1,234.51
GULF LAND USA              DALLAS           7,100.00
TEXAS LIFE & CASUALTY      DALLAS             543.21
CHEMICAL MUTUAL            FORT WORTH         931.72
PALMOLIVE INNS             FORT WORTH         758.93
UNION TRANSPORTATION       GALVESTON            0.00
SOUTHLAND STORES           HOUSTON            342.91

TOTAL                                      10,911.28
```

RDF automatically computes the width of the printed field containing the total for summary and annotated values. The width is computed as the defined field length plus five. You can override this by defining the width on the Detail fill-in.

Obtaining Multiple Values

You can request summary values for one or more fields. For example, assume that the total amount for year-to-date sales (CUSTOMER.YTD) is obtained with the total outstanding amount owed (CUSTOMER.OPEN$). To display these values in summary notation, specify the Detail fill-in as shown in the following screen:

```
=>
 -------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.      RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  ------------------------- - -   - - -  - - - -  - -- --- -------------
 ======  =========== T O P ======= = =   = = =  = = = =  = == === =============
 000400  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  U __ ___ _____________
 000500  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  U __ ___ _____________
 000600  CUSTOMER.STATE             _ _   1 2 N  _ _ _ _  _ __ ___ _____________
 000700  CUSTOMER.OPEN$             _ _   _ _ _  S _ _ _  U __ ___ Z,ZZZ,ZZ9.99_
 000700  CUSTOMER.YTD               _ _   _ _ _  S _ _ _  U __ ___ Z,ZZZ,ZZ9.99_
 ======  ========= B O T T O M === = =   = = =  = = = =  = == === =============
```

The output for the state of Texas displays as the following:

```
Customer Name          City          Open Dollar     Year-To-Date

** STATE TX
AFTON INDUSTRIES       DALLAS           1,234.51        42,394.82
GULF LAND USA          DALLAS           7,100.00        10,452.00
TEXAS LIFE & CASUALTY  DALLAS             543.21           987.32
CHEMICAL MUTUAL        FORT WORTH         931.72         5,231.00
PALMOLIVE INNS         FORT WORTH         758.93         9,789.22
UNION TRANSPORTATION   GALVESTON            0.00           401.22
SOUTHLAND STORES       HOUSTON            342.91         6,349.21

TOTAL                                  10,911.28        75,604.79
```

Notice the keyword TOTAL on the left of the last line in the sample output. The total values for the two columns print under the appropriate headings.

Similarly, you can specify more than one value for each field. If the Detail fill-in from the previous example was modified to specify all of the functions in summary format, the function summary lines for the state of Texas displays as the following:

```
Customer Name          City          Open Dollar     Year-To-Date

UNION TRANSPORTATION   GALVESTON            0.00      401.22
SOUTHLAND STORES       HOUSTON            342.91    6,349.21

TOTAL                                  10,911.28   75,604.79
AVERAGE                                 1,558.75   10,800.68
MINIMUM                                     0.00      401.22
MAXIMUM                                 7,100.00   42,394.82
```

Note: There are times when a function provides the information that is needed without displaying the data in the details. For example, you can expand the previous report to include the number of customers that have no accounts receivable. To do this, a working data variable is created. This variable is a one-digit numeric named NOORDCTR. In the FOR construct, the value of NOORDCTR is set to 0 when the customer has an outstanding balance and to 1 when the customer does not.

```
FOR EACH CUSTOMER
. . .
   IF OPEN$ EQ 0
      SET NOORDCTR = 1
   ELSE
      SET NOORDCTR = 0
   ENDIF
   . . .
   PRODUCE CUSTRPT
ENDFOR
```

On the Detail fill-in, the TOT function is requested as an annotated value for the field NOORDCTR. The field width is set to 0. This prevents the data and any headings from printing with the detail lines.

```
=>
 ------------------------------------------------------------------------------
 IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  ------------------------   - -   - - -  - - - -  - -- --- ------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === ============
 000400  CUSTOMER.NAME              _ _   _ _ _  _ _ _ _  U __ ___ ____________
 000500  CUSTOMER.CITY              _ _   _ _ _  _ _ _ _  U __ ___ ____________
 000600  CUSTOMER.STATE             _ _   1 2 N  _ _ _ _  _ __ ___ ____________
 000700  CUSTOMER.OPEN$             _ _   _ _ _  S _ _ _  U __ ___ Z,ZZZ,ZZ9.99_
 000800  CUSTOMER.YTD               _ _   _ _ _  S _ _ _  U __ ___ Z,ZZZ,ZZ9.99_
 000900  NOORDCTR                   _ _   _ _ _  A _ _ _  U 00 ___ ____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === ============
```

Since a user-defined heading for NOORDCTR is defined on the Column Heading fill-in as CUSTOMERS HAVING NO OUTSTANDING ORDERS:, the following annotated total prints when two customers have no outstanding orders:

```
TOTAL CUSTOMERS HAVING NO OUTSTANDING ORDERS: 2
```

Including Report Functions in Footings

The functions available to obtain values for individual fields are included on the Detail fill-in. The Function column provides for total, average, maximum, and minimum values. By default, the requested values print in summary form or annotated form at each control break level. You can also obtain and print these values in the page or control break footing by specifying them on the Heading fill-in.

A function request on the Heading fill-in overrides any requests for the same information on the Detail fill-in. Any other function specifications on the Detail fill-in are not affected. Similarly, specifications on the Heading fill-in do not have to coordinate with specifications on the Detail fill-in.

For example, a report is to print the total amount outstanding for all customers in each state in the control break footing. CUSTOMER.STATE is defined as a level-1 control break. The total function was not specified on the Detail fill-in. It is specified on the Heading Fill-in to obtain the total value of CUSTOMER.OPEN$, a field defined on the Detail fill-in. The Heading fill-in is defined as shown in the following screen:

```
=>

 -------------------------------------------------------------------------------
 IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal, Function,                   Column
 Command or Arithmetic Expression                         Wid Tab Edit Pattern
 ------  ----------------------------------------------- --  --- -------------
 ======  ==================== T O P ===================== ==  === =============
 000300  <<BF>> LEVEL 1                                   __  ___ _____________
 000400  'Current State:'                                 __  ___ _____________
 000500  STNAMES                                          __  +02 _____________
 000600  'Outstanding Amount:'                            __  040 _____________
 000700  $TOT(CUSTOMER.OPEN$)____________________________ __  +02 _____________
 ======  ===================== B O T T O M ============== ==  === =============
```

When the current level 1 control break is for Texas, the footing prints as the following:

```
Current State:  TEXAS      Outstanding Amount:  10,911.28
```

You can derive values for the current page by specifying the function as all or part of the <<PF>> group. Each page footing contains the requested value for the current page.

If a function is requested on the Detail fill-in and is included in the <<BF>> group on the Heading fill-in, the default control break footing does not print except at the end of the report. The specifications on the Heading fill-in print as part of the report as designated, as summary or annotated values.

Available Report Functions

You can use a number of report functions in report footings. You can also use some report functions in report headings and detail. You can specify more than one function. The value that the function returns is based on those details processed in the specified limit, which is defined by the type of footing: page break by the page footing, <<PF>>, or control break level by the control break footing, <<BF>> with level number.

The following are the report functions that are valid only in footings:

$TOT(field)

Displays the accumulated total value.

$CTOT(field)

Displays the accumulated report total value.

$AVG(field)

Displays the average for details processed.

$MIN(field)

Displays the minimum value encountered.

$MAX(field)

Displays the maximum value encountered.

$OCC(field)

Displays the count of non-null occurrences of field, where field represents the Detail fill-in field name or label. Array elements are not permitted.

The following are functions that are valid in report detail, headings, and footings:

$RPT-DATE

Displays the report date, printed in page heading or footing.

$RPT-PAGE

Displays the page number, printed in page heading or footing.

$OCC

Displays the count of primary PRODUCE statements executed.

Note: If you specify a string function, a WIDTH specification is required.

For example, to obtain the average for accounts receivable (the CUSTOMER.OPEN$ field), specify the function $AVG as the following:

```
$AVG(CUSTOMER.OPEN$)
```

Note: The function $OCC is particularly useful since it can obtain the count of non-null occurrences of a primary or secondary field. For example, a report is defined to print customer information as the primary group level and a list of outstanding orders as a secondary group level. You can use $OCC to obtain a count of the number of outstanding orders for each customer. Assuming that a level-1 control break is defined for customer name and that ORDER.ORDID is a secondary group field containing unique order IDs, the entry in the Heading fill-in for a control break footing to contain the count value is shown in the following screen:

```
=>
 -------------------------------------------------------------------------------
 IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal, Function,                  Column
 Command or Arithmetic Expression                        Wid Tab Edit Pattern
 ------  ----------------------------------------------- --  --- -------------
 ======  ==================== T O P ==================== ==  === =============
 000300  <<BF>> LEVEL 1                                  __  ___ _____________
 000400  'Total Number of Orders:'                       __  ___ _____________
 000500  $OCC(ORDER.ORDID)                               __  +02 _____________
 ======  ===================== B O T T O M ============== ==  === =============
```

Using Labeled Values

You can use labeled values or variables defined in the Detail fill-in in the heading text. See the section titled Including Other Types of Fields earlier in this chapter for information on how to specify labeled values.

For example, in a billing application, a field that contains a labeled expression is defined on the Detail fill-in. This expression is called COST and consists of the result of multiplying the number of units ordered by the cost per unit. The calculation is not coded in the procedure; rather, the field is specified on the Detail fill-in as the following:

```
COST = ITEM.COMMITQ * ITEM.UNITPRICE
```

Although a working data variable is not defined for the label COST, you can use it in the control break footing to print the order total. In other words, you can obtain the value in COST in the facilities of RDF. It is not available procedurally.

To print the total cost for an order, define a level-1 control break for the order ID number, ORDER.ORDID, to cause a control break for each new order. Then define a level-1 control break footing on the Heading fill-in to obtain the total for COST for all items in the order using the $TOT function. This is specified as shown in the following:

```
=>
 -------------------------------------------------------------------------------
 IDEAL: RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST             SYS: DOC  DISP

         Field Name, Literal, Function,                Column
 Command  or Arithmetic Expression                     Wid Tab Edit Pattern
 ------  --------------------------------------------- --  --- -------------
 ======  ==================== T O P ==================== ==  === =============
 000300  <<BF>> LEVEL 1
 000400  'Thank You for Your Order'                    __  ___ _____________
 000500  ______________________________________________ __  L01 _____________
 000600  'Order Total:'                                __  040 _____________
 000700  $TOT(COST)                                    __  +02 _____________
 ======  ===================== B O T T O M ============= ==  === =============
```

You can use the TAB specifications to align the value of $TOT(COST) with the appropriate column in the report details.

The labels for expressions on the Detail fill-in should be unique names, not names defined for any other data (for example, working data, parameter data, columns, and so on). A compile error occurs if a reserved word is used.

Calculating Values in a Footing

You can calculate values in the control break and page footing specifications and include the results with the footing text. The $CALC function performs the calculations.

For example, two columns in the CUSTOMER table, OPEN$ and YTDSALES, can compute the difference between the outstanding balance after processing the current orders and the previous outstanding balance. The calculation is coded for the control break footing on the Heading fill-in. The value prints when a level-1 control break, based on state, occurs. The calculation is specified as:

```
$CALC ( $TOT(CUSTOMER.YTDSALES) - $TOT(CUSTOMER.OPEN$) )
```

When specifying $CALC, you can specify any valid arithmetic expression. The expression can include parenthesis, operators, report functions, field names, and numeric values. All fields and functions must be numeric. Field names must be unique names or labels defined in the primary detail group.

Valid operators include the following:

```
-        *        /        $SQRT
```

Note: You can use $CALC to calculate the standard deviation. For example, to obtain the standard deviation for the value in OPEN$, a field containing the outstanding amount owed for each customer, a labeled detail expression is specified on the Detail fill-in as the following:

```
OPEN2 = OPEN$ ** 2
```

The calculation in the footing on the Heading fill-in is specified as the following:

```
$CALC ($SQRT($OCC(OPEN$) * $TOT(OPEN2) - $TOT(OPEN$) * $TOT(OPEN$))
       / $OCC(OPEN$))
```

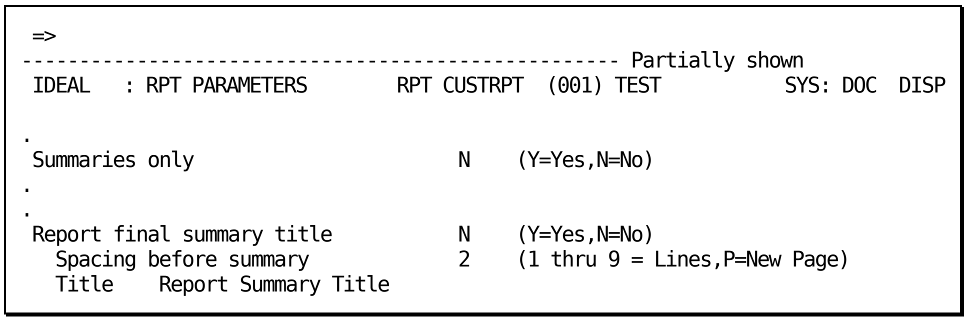
Be sure to define an EDIT PATTERN large enough to hold the largest expected value.

Printing Report Summaries

Summary information consists of the final totals and other data printed at the end of a report. This information is based on the functions specified in the Detail fill-in. It contains the final totals for all control break groups. These values print whether control breaks were specified to print or were specified at all.

For example, in a previous example, the report contained the function values for each state at the control break level. If there were no control breaks, the function values print as the summary at the end of the report.

You can specify a title and some formatting for summary information as shown in the following screen:



```
 =>
---------------------------------------------------- Partially shown
 IDEAL   : RPT PARAMETERS       RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

.
 Summaries only                          N    (Y=Yes,N=No)
.
.
 Report final summary title              N    (Y=Yes,N=No)
   Spacing before summary                2    (1 thru 9 = Lines,P=New Page)
   Title    Report Summary Title
```

You can specify the summary information to begin on a line 1 to 9 lines after the last detail line or on a new page. Specify the title in upper and lower case as it should appear on the report.

Summaries Only

You can use RDF to generate a report containing only the summary information.

On the segment of the previous Parameter fill-in, a prompt is provided for Summaries Only. If you specify yes, detail lines do not print. The report contains all report and page headings and footings, summary title, and summary function values. The annotated count, if requested, prints as part of the summary information.

Generating Multiple Reports

A single application run can generate multiple reports simultaneously. That means more than one report, up to a maximum of 15 reports, can be active at a time. This is useful when the same data is needed to produce several reports.

For example, assume customer reports are printed daily, weekly, monthly, and so on. Each report relies on the same data, but has different RDF specifications. Regardless of these different specifications, the reports can all be produced in a single FOR construct. These reports access all of the customer rows. The following simplified program sample assumes that the subprocedure CHECKDATE sets several flags based on the current date. These flags indicate which reports are generated:

```
FOR EACH CUSTOMER
    PRODUCE DAILYRPT
    DO CHECKDATE
    SELECT EVERY
       WHEN WEEKEND
          PRODUCE WEEKRPT
       WHEN MONTHEND
          PRODUCE MONTHRPT
       WHEN QTREND
          PRODUCE QTREND
       WHEN YREND
          PRODUCE YREND
    ENDSELECT
ENDFOR
```

RELEASE REPORT

The RELEASE REPORT PDL statement closes the report. This also occurs by default at the end of the run of the program. It might be useful to explicitly release a report for one of the following reasons:

- If a report is routed to a network printer from an online program, the RELEASE statement closes the report, scheduling it for printing as soon as the printer is available.
- So that a new report, using the same report definition, can be produced later in the same program run.
- When a program finishes with it since there is a limit of 15 reports that can be simultaneously active.

For the format of the RELEASE statement, see RELEASE Statement in the *Programming Reference Guide*.

Changing Output Destination

You can change the output destination of reports before and during runtime. The ASSIGN command is specified before invoking the application. The ASSIGN statement is used in a program to dynamically change the output destination.

Changing Destination for the Application Run

Use the ASSIGN command to change the destination of a report before running the application. You can issue a separate ASSIGN command for each report generated by the application. In other words, the output destination for each report is distinct from the output destination of the other reports and can be controlled individually.

For example, an application run generates a report called CUSTRPT, which is normally printed on a system printer. For this execution of the application, the report must be sent to a network printer identified as PNP1. Enter the ASSIGN command as follows:

```
ASSIGN REPORT CUSTRPT DESTINATION NETWORK PNP1
```

You can also send the output to CA Email+. For example, the following command sends the output from the report named CUSTRPT to CA Email+, assuming that Jack Smith is a valid CA Email+ ID:

```
ASSIGN REPORT CUSTRPT DESTINATION MAIL 'JACK SMITH'
```

You can also change the output file name for a specific run of a report; however, the CA Ideal JCL must include the new filename. For example, the JCL could include:

```
z/OS:    //IDLRTX   DD SYSOUT=*
VSE:     //DLBL     IDLRTX....
```

Changing the Output File Dynamically

You can assign a report to an output file and destination with an ASSIGN statement during program execution.

You must specify the ASSIGN statement in the application before the first PRODUCE statement executes. Additionally, by using RELEASE and ASSIGN statements during program execution, you can use multiple output files and destinations with a single RDF report definition resulting in multiple reports. You can write the output to a different file and destination for each released report, assuming that the output file names and the destinations are valid.

The ASSIGN statement can specify the output file name as a literal, such as the following:

```
ASSIGN REPORT CUSTRPT TO 'IDLRTX'
```

Or as a variable. If DDNAME is a defined variable, you can use the following statement:

```
ASSIGN REPORT CUSTRPT TO DDNAME
```

Using Working Data

You can use the RELEASE and ASSIGN statements in the program code to split the customer report into different files for each state. The following code segment releases the current report, CUSTRPT, and assigns the output to another file and output destination before obtaining the next set of customers.

The sub procedure GETSTATE uses a table to set the variable STNAME to the name of the state and DDNAME to the name of an output file. The value of STNAME is the base (for example, IDLRTX is generated for Texas). DDNAME corresponds to an output file named in the z/OS JCL or as a VSE DLBL, depending on the operating system.

In this example, the value of STNAME is used in the WHERE criteria on the FOR construct. A flag, NOMORE, is set to true in the subprocedure GETSTATE when the table is exhausted. This terminates the LOOP processing.

```
<<CUST-RPT>> PROCEDURE
   SET NOMORE = FALSE
   LOOP
      DO GETSTATE
   UNTIL NOMORE
      ASSIGN REPORT CUSTRPT TO DDNAME
      FOR EACH CUSTOMER
           WHERE CUSTOMER.STATE EQ STNAME
         MOVE CUSTOMER TO CUST-INQ BY NAME
         PRODUCE CUSTRPT
      WHEN NONE
         LIST $STRING('NO CUSTOMERS IN ',STNAME,'ON FILE')
      ENDFOR
      RELEASE REPORT CUSTRPT
   ENDLOOP
ENDPROC
```

The report is assigned before executing the PRODUCE statement. The report is released when there is no more data for the current report. The next execution of the PRODUCE statement sends the output to the next assigned output file.

Using a Database Table

The ASSIGN statement is executes in the program to change the output destination based on runtime conditions.

For example, assume that an application was created that writes a report CUSTRPT. This application is available to several users at different locations. To ensure that each user's request for the report prints at a location available to the user, the ASSIGN statement is used.

For this example, assume that a table named USERS is defined to contain the user identification number and the identification number of the local printer available to that user. The following is a sample of the data in the table:

USERID	PRINTED
BROWN	PNP1
JONES	PDL2
SMITH	PRS8

When the user requests the report, the program retrieves the printer identification number based on the value of $USER-ID (a function that returns the current user identification number) and sets a working data variable, PRINTID. The program code contains the following:

```
FOR FIRST USERS
     WHERE USERS.USERID EQ $USER-ID
   SET PRINTID = USERS.PRINTID
ENDFOR
ASSIGN REPORT CUSTRPT DESTINATION NETWORK PRINTID
```

Assigning Other Options

You can modify other values for the application run. These values include the following:

- DISPOSITION-KEEP, HOLD or RELEASE
- MAXLINES-Maximum lines per report
- DESCRIPTION-1- to 32-character description
- PAGE SIZE-Maximum number of lines per page
- COPIES-Number of copies to print
- DATE-Report date
- PAGE NUMBER-Starting page number

For more information about the ASSIGN statement, see the *Programming Reference Guide*.

Reporting Exception Conditions

RDF provides the ability to document an exception condition in a report. You can detect an exception condition when no PRODUCE statements were executed for a report or when the value of the return code exceeds a user-specified value. Two group functions are available to handle these conditions:

- <<EMPTY>>-Data following prints when no PRODUCE statements execute for the report.
- <<ABORT>> RC x-Data following prints when the value of the return code is equal to or greater than *x* at the time the report is released (either automatically when the program terminates or by RELEASE REPORT WITH ABORT).

Using <<EMPTY>>

You can specify an <<EMPTY>> group once on the Heading fill-in. You can specify the data to print only as a literal. It prints when no PRODUCE statements execute for the report. The data can be formatted through the tab field to print on multiple lines. If you do not specify an <<EMPTY>> group, there is no output when PRODUCE statements do not execute for the report.

For example, you can specify the following <<EMPTY>> group:

```
=>
 -------------------------------------------------------------------------------
 IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST        SYS: DOC  DISP

         Field Name, Literal, Function,                   Column
 Command or Arithmetic Expression                         Wid Tab Edit Pattern
 ------  ------------------------------------------------ --  --- -------------
 ======  ==================== T O P ===================== ==  === =============
 000300  <<EMPTY>>                                        __  ___ _____________
 000400  'There is no output data for this report.'       __  ___ _____________
 000500  ________________________________________________ __  L01 _____________
 000600  'Check for appropriate database.'                __  ___ _____________
 ======  ===================== B O T T O M ============== ==  === =============
```

Prints as:

```
There is no output data for this report.
 Check for appropriate database.
```

Using <<ABORT>>

You can specify an <<ABORT>> group once on the Heading fill-in. You can specify the data to print as a literal, variable (non-sorted reports), or expression. It can be formatted using the tab field to print on multiple lines. If you do not specify an <<ABORT>> group, the final footings are suppressed and a standard CA Ideal supplied message prints at the end of the report.

For example, the following <<ABORT>> group specifies to print when the return code is 12 or greater than 12.

```
=>
 -------------------------------------------------------------------------------
 IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal, Function,                Column
 Command or Arithmetic Expression                      Wid Tab Edit Pattern
 ------  ----------------------------------------------- --  --- ------------
 ======  ==================== T O P ===================== ==  === ============
 000300  <<ABORT>> RC 12                                 __  ___ ____________
 000400  'A fatal error has occurred.'                   __  ___ ____________
 000500  _______________________________________________ __  L01 ____________
 000600  'The return code is:'                           __  ___ ____________
 000700  $RC                                             __  +02 99__________
 ======  ===================== B O T T O M ============== ==  === ============
```

When the return code is set to 12 or greater, the output contains the following:

```
A fatal error has occurred.
The return code is:  12
```

When the report is released implicitly by program termination, the appropriate <<ABORT>> text prints.

You can use the <<ABORT>> group with user-defined error procedures. Although you can code only one <<ABORT>> group, you can specify one or more working data variables to define the output, provided that the variables are current for the last PRODUCE REPORT statement executed. Variables set after the last PRODUCE REPORT statement was executed are not available for the <<ABORT>> group. For sorted reports, the variable is not available for the abort group.

When the return code is less than 12, a user-written Error Procedure can often recover from the error. The program is not terminated, but you can explicitly terminate the report using the RELEASE statement:

RELEASE REPORT *name* WITH ABORT

Any specified <<ABORT>> text for that return code then prints to provide diagnostic or informational messages to the user. The program can resume processing. The resulting report prints using the same specifications as the previously released report.

Chapter 3: Checking the Report Definition

Once a report is defined in CA Ideal, you can review the definition online or on paper. You can also check your report definition by producing a prototype of the report.

Displaying the Report Definition

To review the definition of a report online, enter the command DISPLAY REPORT. You are prompted for the name of the report. You can enter the complete command to avoid the prompter. See the *Command Reference Guide* for complete syntax.

When you display a report definition, the Detail fill-in is displayed by default. To review information entered on other report definition fill-ins, you can either press the appropriate PF key or enter the name of the fill-in. You can abbreviate the name to the standard three characters or an alternate):

Name	**Abbreviation**	**Function Key**
COLUMN	COL	F6
DETAIL	DET	F6
HEADING	HEA	F5
IDENTIFICATION	IDE	
PARAMETERS	PAR, PARM	F4

Printing the Report Definition

You might want to keep a hard copy of the report definition as part of the program documentation. To print a copy of the report definition, use the command PRINT REPORT. You are prompted for the name of the report. You can enter the complete command to avoid the prompter. See the *Command Reference Guide* for complete syntax.

When you print a report definition, the definition is shown as screen images of the fill-ins you completed online. Each fill-in is included in the printout, regardless of whether the fill-in was used in the definition.

Prototyping the Report

Prototyping is a convenient way to test a report format before actually coding, compiling, and testing the complete program that generates the report. CA Ideal generates random values for the primary group of the detail definition as sample data for the report. This is normally sufficient to check alignment, spacing, pagination, and edit patterns.

Only the primary group of the report definition is produced in the report prototype. In addition, each field produced for the report prototype contains, at most, 52 characters, regardless of the actual field length defined for the report.

Prototyping Prerequisites

To prototype a report, you must complete the report definition and define a skeleton program.

The report definition must include:

- At least one variable field in the primary group from working data or a dataview field.
- If labels designate primary and secondary report groups, the first line of the report detail definition must contain the primary group label.

The program must include:

- The report as a resource
- All report fields identified as a dataview column, a working data variable, or a parameter data variable
- A FOR construct for every dataview referenced in the report. Since logic is not required, you can code the program as:

```
<<SAMPLE>> PROCEDURE
   FOR EACH dataview1
   ENDFOR
   FOR EACH dataview2
   ENDFOR
ENDPROC
```

The program must compile successfully.

PRODUCE Command

The PRODUCE REPORT command produces the prototype using arbitrarily generated data. For example, assume that a program named CUST contains a report TEXCUST that lists all of the customers in Texas. To obtain a prototype of the report, specify:

```
PRODUCE REPORT TEXCUST FOR PROGRAM CUST DETAIL 50
```

A prototype is generated for the report named TEXCUST in the program CUST. The Detail option limits the number of records produced to fifty. The PRODUCE REPORT command is valid for TEST status programs only.

Chapter 4: Generating Reports Procedurally

You can create reports procedurally by relying on the LIST statement to generate the output. You can specify the output using a literal, panel, or one or more variables. The LIST statement for generating output is easy to use. Each time the LIST statement is encountered, output is sent to the RUNLIST file, which can be viewed online using CA Ideal when the program terminates. You can then store or print the output. This is controlled separately from the program that generated the output.

Listing All Rows

You can write the contents of one or more columns from a table to the RUNLIST file by specifying the individual column names with the LIST command. For example, the following code writes a record on the RUNLIST file for each row accessed. Each record contains the contents of the two columns from the CUSTOMER table, CUSTID and NAME:

```
FOR EACH CUSTOMER
    LIST CUSTOMER.CUSTID CUSTOMER.NAME
ENDFOR
```

The report does not have any headings or footings. There is no control over pagination. Since the RUNLIST file is shared by all programs in an application run unit, all reports generated using RUNLIST are contained in the file with any termination or error messages produced using the LIST statement.

Creating Page Breaks

The LIST NEWPAGE statement sends a page eject request to the RUNLIST file. When the file prints, the page ejects force the following records to print on a new page. For example, the following code causes a page break before the customer data begins to print and after the customer data finishes printing:

```
LIST NEWPAGE
FOR EACH CUSTOMER
    LIST CUSTOMER.CUSTID CUSTOMER.NAME
ENDFOR
LIST NEWPAGE
```

You can use this technique to separate reports generated by the LIST statement in the same application run.

Controlling Page Breaks in a Report

The printer setup automatically controls the number of lines printed on the page. If a different specification is necessary, it must be handled procedurally. That means the procedure must count the number of lines generated (that is, the number of LIST statements) and specify LIST NEWPAGE as needed to force pagination.

```
SET LINE-CTR = 1
LIST NEWPAGE
FOR EACH CUSTOMER
   IF LINE-CTR > 55
       LIST NEWPAGE
       SET LINE-CTR = 1
   ENDIF
   LIST CUSTOMER.CUSTID CUSTOMER.NAME
   SET LINE-CTR = LINE-CTR + 1
ENDFOR
LIST NEWPAGE
```

This example limits the output to 55 lines per page. Notice that a working data variable, LINE-CTR, was defined to act as a counter.

Listing Several Reports in One Application Run

The LIST NEWPAGE statement paginates a single report and separates multiple reports in a single application run. However, the LIST NEWPAGE statement maintains all of the data in one file. To separate several reports using the same RUNLIST file into distinct files, you can release the file for printing between reports. The RELEASE REPORT RUNLIST command can close the RUNLIST file and release the current contents for printing. A new RUNLIST file is opened before returning control to the application.

Since there is only one output file active at a time, multiple reports cannot be generated simultaneously. They must be produced one at a time.

Formatting the Report

You can format reports created with the LIST statement in two ways, using procedural code or using panels.

Formatting Procedurally

To handle formatting procedurally, you must specify the data exactly as it is to print. Therefore, you must include spaces between columns when specifying the columns to print. The heading, provided as a literal or as one or more variables, must also contain spaces for the proper column alignment:

```
DO PRT-HEAD
FOR EACH CUSTOMER
    IF LINE-CTR > 55
       DO PRT-HEAD
    ENDIF
    LIST CUSTOMER.CUSTID,'     ',CUSTOMER.NAME
    SET LINE-CTR = LINE-CTR + 1
ENDFOR
LIST NEWPAGE
  . . .
<<PRT-HEAD>> PROCEDURE
   LIST NEWPAGE
   LIST 'ID         Name'
   LIST SKIP
   SET LINE-CTR = 3
ENDPROC
```

Notice the use of LIST SKIP in the PRT-HEAD procedure to force a blank line between the column headings and the detail lines. You can insert multiple lines by specifying SKIP as many times as the number of lines you want to skip. For example, to skip three lines specify:

```
LIST SKIP SKIP SKIP
```

Formatting with Panels

Panels provide a powerful means of formatting the output when using the LIST statement. The format is designed on the panel, then the data is moved to the panel before the LIST statement.

For example, if a panel named CUSTPNL was defined to contain a specific number of repeating detail lines to fit on the page, you can produce a report with the following code:

```
SET PNL-IDX = 1
FOR FIRST 10 CUSTOMER
    MOVE CUSTOMER TO CUSTPNL.CUSTINFO(PNL-IDX)
    SET PNL-IDX = PNL-IDX + 1
ENDFOR
LIST PANEL CUSTPNL
```

The panel is first populated in the FOR construct and then listed outside the FOR construct.

Using LIST for Preprinted Forms

Using a panel to format LIST output is particularly useful for printing data on preprinted forms. The panel is created with the exact format of the preprinted form, including the literal text, headings, and footings. The literal data can print during testing to provide a prototype of the actual form and then for production, the literal data on the panel can be designated as invisible (or nonprinting) or can be deleted from the panel.

Although you can specify a panel to contain up to 236 characters on a single line, the LIST file maximum length is 132 characters. Data is truncated if it extends beyond the LIST file limit.

Customer Inquiry: An Example

The following example demonstrates using LIST to generate information on a preprinted form.

Assume that a panel named CUSTPNL was created that corresponds to the form. The panel definition input parameters do not specify error handling, value limitations, and so on, since there is no user interaction. The output parameters specify the necessary edit patterns. All fields are protected and, when testing, all fields display.

The following code generates a report for all customers in Texas.

```
<<CUST-RPT>> PROCEDURE
    FOR FIRST CUSTOMER
        WHERE CUSTOMER.STATE EQ 'TX'
        MOVE CUSTOMER TO CUST-INQ BY NAME
        LIST PANEL CUST-INQ
        LIST NEWPAGE
    WHEN NONE
      LIST 'THERE ARE NO CUSTOMERS IN TEXAS ON FILE'
    ENDFOR
ENDPROC
```

While testing the program, you can view the output online to evaluate the report content quickly. You can print the output to verify the layout.

When the program is ready to be released to production, you can remove the headings and literals that simulate the preprinted form from the panel. If you want to maintain these literals for documentation or future updates, you can retain them by designating them as nonprint by specifying a field attribute of I (for invisible) on the Extended Field Definition fill-in of the Panel Definition Facility.

Printing and Prototyping Reports

The LIST PANEL statement prints the contents of a panel in the RUNLIST output of an application. Each time a LIST PANEL is issued, a new page is generated in the RUNLIST report.

The original purpose of the LIST PANEL statement was to provide a debugging and documentation tool for online applications. You can use the statement in an error procedure to print one or more panels accessed by the program that caused an error to occur.

You can also use the LIST PANEL statement to create a proof of an online program by documenting the online inputs and (with different LIST statements) detailing the program's outputs.

Another way to use the LIST PANEL statement is to print reports. The advantages to using this method are:

1. You can prototype the report layout online using the same high-productivity techniques available for prototyping user panels. Using the layout and facsimile features of panel editing, you can quickly lay out the format of a report and display it to the user for approval.

   If your report width goes beyond 80 characters, you must use the following command before editing the panel parameter:

   ```
   SET PANEL WIDEOPTION YES
   ```

   Change the panel width option on the Parameter fill-in to a value up to 132 characters, as needed for your application. The RUNLIST file is always allocated with a width of 132 bytes, although various physical devices can print reports wider than 132 bytes.

   You can scroll in both layout and facsimile modes to show how the report looks. You can set repeated detail lines in the report in a Panel Summary definition as a repeating group with either fixed or variable occurrences.

2. By naming fields in the panel summary identically with database or program-developed fields, you can use MOVE/SET BY NAME to minimize the amount of code used in loading panel data during execution.

3. LIST PANEL supports the invisible attribute (not printed) and the highlight attribute (double-print for normal highlighting, extended highlighting features are not supported in the printout).

4. This method is particularly well-suited for preprinted forms, with totals printed in fixed positions on the form.

Some additional considerations are:

1. You can print only one report per application. This is because the LIST PANEL statement uses the application's single RUNLIST file that is shared by the main program and all subprograms in an application run-unit.

2. LIST statements coded for debugging and in error procedures that are left in the program can interfere with proper formatting of a report by adding unwanted information at the bottom of each form. Do not use this technique in complex applications because the main program or any of its subprograms can add this information.

3. Using the LIST PANEL technique of printing reports truncates any data beyond the 132-byte width.
4. The advanced manipulation techniques of the CA Ideal Report Writer are not available with LIST PANEL, therefore:
   - Sorted reports and control breaks must be specifically provided for your programs.
   - Automatic functions such as TOTAL, AVERAGE, MAXIMUM, and MINIMUM, must be coded explicitly into your program.
   - ASSIGN statements and commands that apply to report entities do not apply to LIST PANEL reports.
   - Date data types must be explicitly translated into user-specified date formats since panels do not support the date data type.
   - This technique does incur more overhead than using the Report Writer, so it might be appropriate only for producing relatively small amounts of output.

Providing Summary Information

If you want to include summary information when using LIST, you must include code in the program, you must define variables to accumulate the values, and you must list the content of those variables with any formatting for the value. Actually, it is more efficient and easier to use RDF if the report contains subtotals for groups of detail lines and totals on a final page. RDF also generates reports that contain ONLY the summary information.

Listing Several Reports on One Page

Since LIST PANEL does not automatically cause pagination, several logical reports can be generated on one physical page. For example, assume a customer inquiry report should contain information about all customers in Texas. The inquiry is actually in two parts-first, general information from the CUSTOMER table and, second, a list of outstanding orders, if any.

Two panels are created to correspond to a preprinted form: The CUST-INQ panel displays the general customer information and the ORD-INQ panel displays a list of orders. Since a repeating group is used in the ORD-INQ panel to display the order list, the panel is populated before it is listed. You do not need to list the panels in the same program. In fact, the ORD-INQ panel is listed from a subprogram called GETORD. To specify which orders are retrieved, the CUSTOMER.CUSTID is passed as a parameter to GETORD. Pagination (that is, LIST NEWPAGE) occurs in the main program to force a new page for each customer:

```
<<CUST-RPT>> PROCEDURE
    FOR EACH CUSTOMER
        WHERE CUSTOMER.STATE EQ 'TX'
        MOVE CUSTOMER TO CUST-INQ BY NAME
        LIST PANEL CUST-INQ
        CALL GETORD INPUT CUSTOMER.CUSTID
        LIST NEWPAGE
    WHEN NONE
      LIST 'THERE ARE NO CUSTOMERS IN TEXAS ON FILE'
    ENDFOR
ENDPROC
```

The GETORD program contains the following code:

```
<<ORD-RPT>> PROCEDURE
    SET PNL-IDX = 1
    FOR FIRST 20 ORDER
        WHERE ORDER.CUSTID EQ CUSTOMER.CUSTID
        MOVE ORDER TO ORD-INQ.ORDINF(PNL-IDX) BY NAME
        SET PNL-IDX = PNL-IDX + 1
    WHEN NONE
        SET ORD-INQ.MSG = 'NO ORDERS'
    ENDFOR
    LIST PANEL ORD-INQ
```

The subprogram GETORD limits the number of orders to the size of a fixed length panel.

Controlling the Report Destination

You can assign the RUNLIST file to an output file and destination other than the default. You must include the output file must as a ddname in the JCL and the destination must be valid. To change the output file or destination, specify the ASSIGN REPORT statement before the first LIST statement execution in the application. For example, to assign the RUNLIST file to the file NEWNAME included as a ddname in the JCL, specify:

```
ASSIGN REPORT RUNLIST TO NEWNAME
```

The default destination for the RUNLIST file is the output library. You can change it. For example, the following statement sends the output to CA Email+:

```
ASSIGN REPORT RUNLIST DEST MAIL email-id
```

You can modify other values for the application run. These values include:

- **DISPOSITION** KEEP, HOLD, or RELEASE
- **MAXLINES** The maximum lines per report
- **DESCRIPTION** A 1- to 32-character description
- **PAGE SIZE** The maximum number of lines per page

Note: Although you can specify DISPOSITION, CA Ideal ignores the specification. The specification is there only for mainframe CA Ideal compatibility.

See the *Programming Reference Guide* for detailed information about the ASSIGN statement.

Using LIST Instead of RDF

LIST is useful for producing bare information quickly, for single line reports, and for producing reports on printers that require printer control characters such as channel skips in the first position of the line. RDF does not support such printers. However, you should use RDF to generate most reports since it provides automatic formatting and summary facilities that are unavailable using the LIST statement. In addition, although LIST PANEL does provide a way of formatting LIST output, it is not as efficient as generating formatted output using an RDF report definition.

Appendix A: Fill-In Descriptions

The components of a CA Ideal report definition include the following:

- A report identification fill-in that creates a report definition and provides it with identification information.
- Optionally, a report parameters fill-in that establishes general report options such as width, depth, justification, page numbering, and date.
- Optionally, a report heading and footing definition fill-in that defines headings and footings.
- A report detail definition fill-in that specifies information about detail lines, such as sorting, control breaks, and summary functions.
- Optionally, a report column heading fill-in that specifies column headings to replace defaults that might be defined in the report parameters fill-in.

To move about between one fill-in and another, you must always enter a command to indicate where you want to go. There are two ways to enter these destination commands:

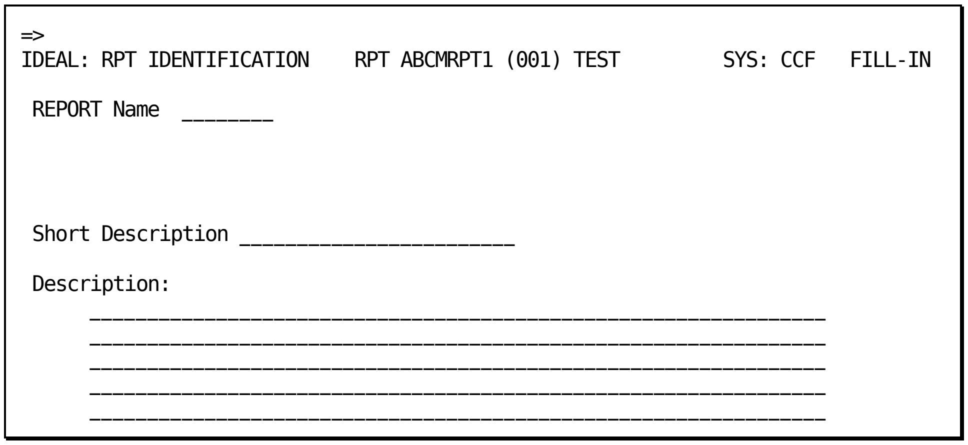
- You can enter the name of the fill-in on the command line at the top of the screen. If you want to return to the menu, enter the command RETURN.
- You can press the PF key assigned to that fill-in or command. See the chart in the section titled *PF key Assignments* in Chapter 2.

If the fill-in was never completed for the current report, it displays with default values where appropriate. If information was previously entered on the fill-in, it displays as it was entered. When you complete the fill-in, press the Enter key to apply the data and leave the current fill-in displayed. Pressing one of the PF keys to display another fill-in applies the changes without redisplaying the fill-in.

Each of the following sections contains examples of fill-ins with default values, if any, as they appear to the user.

Report Identification Fillin

The report identification fill-in enters descriptive information about the report when the report definition is initially created or when it is subsequently modified. The Identification fill-in automatically displays when a report is created.



```
=>
IDEAL: RPT IDENTIFICATION    RPT ABCMRPT1 (001) TEST         SYS: CCF   FILL-IN

 REPORT Name  ________



 Short Description ________________________

 Description:
      ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
```

Enter the following information on the Identification fill-in:

Report name

Defines the one- to eight-character name assigned to the report definition. If a name was assigned in the CREATE REPORT command, that name displays in this field.

Note: RUNLIST is a reserved word for a report name.

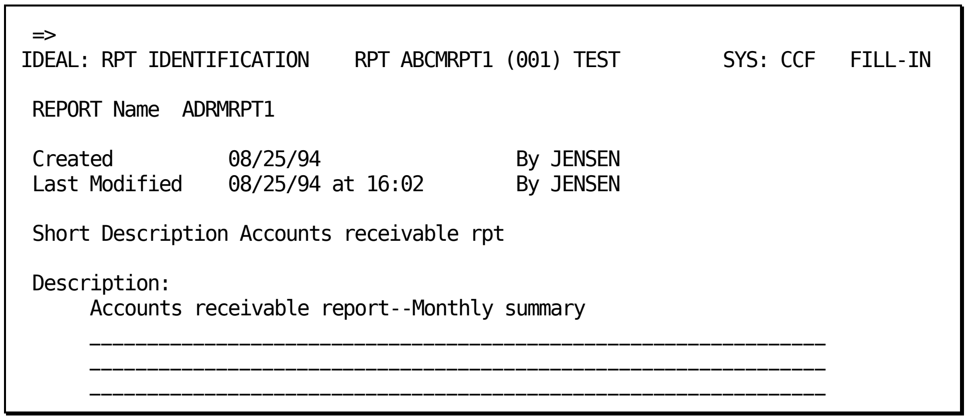
Short description

Displays a brief description of the report definition. This description is limited to 24 characters.

Description (Optional)

Displays a longer description of the report definition. This description is limited to five lines.

The following graphic shows a completed report identification fill-in:



```
 =>
IDEAL: RPT IDENTIFICATION    RPT ABCMRPT1 (001) TEST        SYS: CCF   FILL-IN

 REPORT Name  ADRMRPT1

 Created          08/25/94                 By JENSEN
 Last Modified    08/25/94 at 16:02        By JENSEN

 Short Description Accounts receivable rpt

 Description:
      Accounts receivable report--Monthly summary
      ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
```

The two lines, Created and Last Modified, display information once the Identification fill-in is entered. The system supplies the information in these fields. The user cannot modify it.

Created ... By

Displays initial creation date of the report definition and the user ID of the creator.

Last Modified ... at ... By

Displays date, time, and user ID of the last edit access. Until the report definition is edited, this field contains the original creation date, time, and user ID. The system supplies this information. You cannot change it.

Report Parameters Fillin

Report parameters are specified on the report parameter fill-in. It is an optional form that selects general report layout options such as a report's length and width on a page, the spacing between lines and columns, column headings, control breaks, heading definition, summary information, and date and page specification. If this fill-in is not completed, default values are used for any required parameters. There are three ways to display the report parameters fill-in, as follows:

To display the parameters fill-in for a report that is not the current entity, enter

```
DISPLAY REPORT name PARAMETER
```

on the command line

To display the parameters fill-in for the current report, either enter the command PARAMETER on the command line, or press the F4 key.

The following screen shows the Parameters fill-in.

```
 =>
--------------------------------------
 IDEAL   : RPT PARAMETERS       RPT CUSTRPT  (001) TEST      SYS: DOC  FILL-IN
 Lines per page on printout            ___  (0 thru 250)
 Report width                          ___  (40 thru 230)
Spacing between lines                  _    (1 thru 3)
Spacing between columns                __   (0 thru 66 OR A=Automatic)
Spacing after page and column hdgs     _    (0 thru 9)
Summaries only                         _    (Y=Yes,N=No)
 Column headings desired               _    (Y=Yes,N=No)
 Column headings indication            _    (U=Underscore,N=None,D=Dashes)
Control break heading                  _    (Y=Yes,N=No)
 Control break footing                 _    (Y=Yes,N=No)
 Automatic page footing summaries      _    (Y=Yes,N=No)
 Group continuation at top of page     _    (Y=Yes,N=No)
 Annotated count in break footings     _    (Y=Yes,N=No)
 Report final summary title            _    (Y=Yes,N=No)
   Spacing before summary              _    (1 thru 9 = Lines,P=New Page)
   Title    __________________________________________
 Date
   Position            __   (NO=None, BR=Bot.Right, BL=Bot.Left, BC=Bot.Ctr.,
                             TR=Top Right, TL=Top Left, TC=Top Center)
   Format              MM/DD/YY
 Page Numbers
   Position            __
   Format              _    (D=Digits Only, H=With Hyphens, P= Page      nnn)
Page Heading
   Heading             __________________________________________
   Position            _    (C=Center, L=Left Justify, R=Right Justify)
```

Enter the following information on the Parameter fill-in:

Lines per page on printout

Specify number from 0 to 250 that indicates the total number of lines for each page of the report. This includes all headings, footings, details, and blank lines. Specify 0 to suppress page breaks.

The Lines per page parameter determines when a page break occurs. RDF first determines the number of lines required for the page after the page heading, column headings, page footing, and associated spacing. This value is subtracted from the Lines Per Page specification. The remaining lines are available for the page body, which consists of the details and the control break headings and footings.

If a complete detail does not fit on the page, a page break occurs. When a detail that causes a control break is produced, a page break occurs if the appropriate control break heading and footing do not fit on the page. In this case, the detail and the footing for the current control break group print on the next page.

You can set a session default for this parameter with the command SET RPT LINES. If a default was set, it appears on the Parameters fill-in when it first displays. You can change the default by typing over the displayed value.

Report width

A number from 40 to 230 that specifies the number of characters per line.

You can set a session default for this parameter with the command SET RPT WIDTH. If a default was set, it appears on the Parameters fill-in when it first displays. You can change the default by typing over the displayed value.

Spacing between lines

A number from 1 to 3 that specifies the number of blank lines left between printed detail lines:

- Single spacing, no blank lines.
- Double spacing, one blank line.
- Triple spacing, two blank lines.

You can set a session default for this parameter with the command SET RPT SPACING. If a default was set, it appears on the Parameters fill-in when it first displays. You can change the default by typing over the displayed value.

Spacing between columns

A number from 0 to 66 or the code A. This value specifies the number of blank characters left between columns on detail lines. A number indicates the exact number of spaces between columns. A value of 0 means that columns are concatenated.

The value A indicates that spacing is automatic. The total number of spaces left after subtracting all column widths from the report width is proportionally distributed among the columns and the right and left margins. When you use automatic spacing, if a column does not fit on a line, a warning message occurs during the compile.

You can set a session default for this parameter with the command SET RPT GAP. If a default is set, it appears on the Parameters fill-in when it first displays. You can change the default by typing over the displayed value.

Spacing after page and column headings

A number from 0 to 9 that specifies the number of blank lines after the page heading detail line, column heading detail line, or continuation line and before the first body line of the report.

Note: You must specify a minimum value of 1 in the REPORT PARAMETER under "spacing after page and column headings" to generate the correct carriage control for reports that do not have any DETAIL lines and, therefore, no COLUMN headings and no PAGE headings defined.

A value of 1 for one blank line is the default. To change the default value, type a new value over the current value.

Summaries only

Specifies whether detail lines print:

- **Y** Only the summary lines at each control break print.
- **N** Detail lines print (default).

Column headings desired

Specifies whether column headings print:

- **Y** Defined column headings are included in the report (default).
- **N** Column headings do not print.

Column heading indication

Specifies how column headings are highlighted:

- **U** Column headings are underscored.
- **D** A series of dashes print on a separate line beneath the column heading (default).
- **N** No highlighting.

Control break heading

Specifies whether standard headings print when a control break occurs. The default control break heading consists of asterisks followed by the control field column heading and the control field value. Two asterisks are added for each level of the control break. A level-1 control break is preceded by two asterisks. A level-2 control break is preceded by four asterisks, and so on.

- **Y** Control break headings are printed at the beginning of each control group.
- **N** Suppresses the printing of default control break headings. User-specified control break headings are not suppressed.

Control break footing

Specifies whether standard footings print when a control break occurs. The default control break footing consists of the control field column heading and the control field value.

- **Y** Control break footings are printed at the end of a control group.
- **N** Suppresses the printing of control group footings.

Automatic page footing summaries

Specifies whether summary and annotated value functions (TOT, MIN, MAX, AVG) in Detail fill-in print in the page footing. The page footing contains the requested values for only the current page.

- **Y** Summary and annotated lines print as part of the page footing on the line below the footing title.
- **N** Summary and annotated lines do not print as part of the page footing (default).

Group continuation at top of page

Specifies whether control headings print at the top of a new page when a control break did not occur.

- **Y** Control headings, if any, print at the top of the page followed by (CONT.). The continued indicator (CONT.) prints for every line of the control heading. However, if the control heading uses the entire width of the line, the continued indicator is truncated. Control field values, if any, print at page breaks, regardless of whether the value changed.
- **N** Control fields do not print after a page break when there is no change in the control value.

If control breaks are set for the fields STATE (level 1) and CITY (level 2) and a page break occurs in the middle of details from San Francisco, California, the following headings appear at the top of the page when Group Continuation is set to Y:

```
** STATE CA (CONT.)
**** CITY SAN FRANCISCO (CONT.)
```

If Group Continuation is set to N, the next detail line appears at the top of the page with no control break heading.

Annotated count in break footings

Specifies whether to print an annotated count in standard footings. This is a count of the number of primary detail groups generated from executing the PDL PRODUCE statement since the last control break at the same level. Control footings affected by this parameter are control break footings, page footings, and report footings. User-defined footings are unaffected.

- **Y** Annotated count appears (default).
- **N** Count does not appear.

Report final summary title

Specifies whether to print a final summary line (equivalent to control break footing level 0) when summary functions are specified for a detail field.

- **Y** Final summary (grand total) line prints (default).
- **N** Final summary line does not print.

Note: A value of Y can generate blank lines if no count or summaries are accumulated.

Spacing before summary

- **1-9** Specifies how many lines are skipped before the report final summary title. The default is 2, which skips 2 lines.
- **P** Specifies that the final summary title prints on a new page.

If you specify a larger value for the first line of the LEVEL 0 control break in the Heading fill-in, the spacing specified in the Heading fill-in overrides that specified in the Parameters fill-in.

Title

A title for the final summary control break footing, which appears on a separate line at the end of the report preceding the final summary information. If nothing is specified here, the default is FINAL TOTAL.

Date

Specifies the position and format in which the date prints.

Position

Specifies the location of the date on the page. You can set a default for this parameter with the SET RPT DATEPOS command. Any of the following are valid:

To place the date in	Specify	Position
The page footer	BR	Bottom right
	BL	Bottom left
	BC	Bottom center
The page heading	TR	Top right
	TL	Top left
	TC	Top center
To suppress	NO	None

If you specify NO here, you can still place the date in a user-defined heading or footing specified on the Heading fill-in.

Note: You cannot specify the same position for page number, date, and heading if all are requested. If you also specify a page heading or footing on the Heading fill-in, leave this position blank in the first line of that page heading or footing.

Format

Specifies the format of the date. You can use any logical combination of the date patterns shown in the following table to specify a date format.

Note: You can specify the $RPT-DATE function as a field name on the Heading and Detail fill-ins to produce the report date at another location. You must specify a format for the date here when you use $RPT-DATE. However, you can set a default for this parameter with the command SET RPT DATEFOR. This default date format applies only to the date in the page heading, page footing, and $RPT-DATE. The default format for dates in the Detail fill-in is controlled by the command SET CMD DATEFOR.

The following specifications give the corresponding results when a report is produced, assuming that, at the time of production, the date is January 12, 1993.

Component Notation	Meaning	Example assuming (January 12, 1993)
YEAR	Year in full	1993
YY	Year without century	93
Y	Year without decade	6
MONTH	Month spelled out (upper case)	JANUARY
LCMONTH	Month spelled out (initial letter uppercase)	January
MON	Month abbreviation (uppercase)	JAN
LCMON	Month abbreviation (initial letter uppercase)	Jan
MM	Month number, with leading zero if necessary	01
M	Month number with no leading zero	1
DD	Day with leading zero if necessary	12
D	Day with no leading zero	12
DDD	Julian day, numeric day of the year (1-366)	012
WEEKDAY	Day spelled out (uppercase)	SUNDAY
LCWEEKDAY	Day spelled out (initial letter uppercase)	Sunday

Component Notation	Meaning	Example assuming (January 12, 1993)
DAY	Day abbreviation	SUN
LCDAY	Day abbreviation (initial letter uppercase)	Sun

The following are examples of completed date format specifications.

```
Specification          Result
MM/DD/YY               01/02/93
M/D/YY                 1/2/93
DD-MM-YY               02-01-93
DD/MM/YY               02/01/93
DD MON YEAR            02 JAN 1993
```

Any characters except uppercase alphabetic in the date pattern remain unchanged.

The site administrator defines the actual text indicated by the keywords MONTH, LCMONTH, MON, LCMON, WEEKDAY, LCWEEKDAY, DAY, and LCDAY in the file DATETBL.IDL. For more information about defining this text, see the *Working in the Environment Guide*.

Page numbers

Specifies the location and format in which the page number prints.

Position

Specifies the location of the page number on the page. You can set a default for this parameter with the SET RPT PAGEPOS command. Any of the following are valid:

To place the date in	Specify	Position
The page footer	BR	Bottom right
	BL	Bottom left
	BC	Bottom center
The page heading	TR	Top right
	TL	Top left
	TC	Top center
To suppress	NO	None

If you specify NO here, you can still place the page number in a user-defined heading or footing, specified on the Heading fill-in.

Note: You cannot specify the same position for page number, date, and heading if all are requested. If you also specify a page heading or footing on the Heading fill-in, leave this position blank in the first line of that page heading or footing.

Format

Specifies the format of the page number.

- **D** Digits only; for example, 15
- **H** Hyphens; for example, -15-
- **P** Digits with a one- to eight-character prefix defined by the user. PAGE is the default prefix if the user does not specify one.

You can set the default for the page number with the command SET RPT PAGEFMT. You must specify the page format on the Parameter fill-in or by default if you use the $RPT-PAGE function on the Heading or Detail fill-in.

Page heading

Specifies the literal to use as the page heading and its position on the page.

You should not define both a heading report parameter and a report page heading definition. If you specify date and page as report parameters, the first line of a multiple line heading should be positioned so that it does not conflict with the date and page positions.

Heading

Specifies the 1 to 42-character literal that appears as a page heading at the top of every page of the report. When this field is blank (the default), no literal text appears, even if page numbers or the date are specified using the Parameter fill-in.

Position

Specifies the location of the page heading as:

- **C** Centered (default).
- **L** Flush left margin.
- **R** Flush right margin.

Heading/Footing Definition Fillin

The report Heading/Footing definition fill-in specifies a more elaborate report page heading or footing than can be defined on the report Parameters fill-in. The Heading/Footing fill-in is optional and is shown in the following screen.

Definitions specified with the Heading/Footing fill-in override those specified on the report Parameters fill-in. For example, the standard control break heading parameter on the report Parameters fill-in is ignored for a break level if a <<BH>> label is specified on the Heading/Footing fill-in.

You can use field names, literals, functions, and arithmetic expressions as part of a heading or footing. The position or location of the heading or footing is also specified on this fill-in.

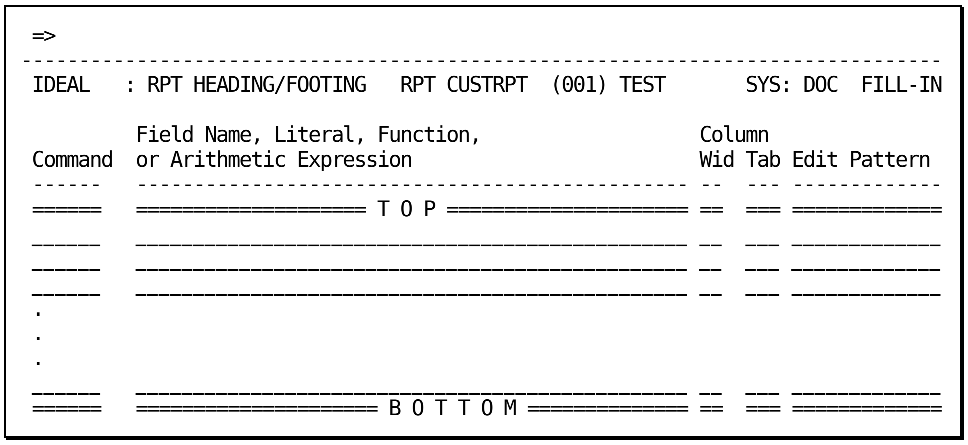
You can define the following types of headings and footings on the Heading fill-in:

- A report heading, identified by the group label <<RH>>, prints once at the beginning of the report on a separate cover page.

  Note: You should not define both a heading report parameter and a report page heading definition. If you specify date and page as report parameters, position the first line of a multiple line heading so that it does not conflict with the date and page positions.

- A report footing, identified by the group label <<RF>>, prints once at the end of the report. It can print on the same page as the last detail or on a separate page.
- A page heading, identified by the group label <<PH>>, prints at the top of every page.
- A page footing, identified by the group label <<PF>>, prints at the bottom of every page.
- One or more control break headings, identified by the group label <<BH>> LEVEL *n*, prints before the first detail of a control break group. Control break headings specified here override the default headings.

- One or more control break footings, identified by the group label <<BF>> LEVEL *n*, prints after the last detail of a control break group. Control break footings specified here override the default footings.
- To display the report Heading fill-in, enter the command HEADING on the command line or press the F5 key.



```
=>
--------------------------------------------------------------------------------
IDEAL   : RPT HEADING/FOOTING   RPT CUSTRPT  (001) TEST       SYS: DOC  FILL-IN

         Field Name, Literal, Function,                 Column
Command  or Arithmetic Expression                       Wid Tab Edit Pattern
------   ------------------------------------------------ --  --- -------------
======   ==================== T O P ===================== ==  === =============
______   ________________________________________________ __  ___ _____________
______   ________________________________________________ __  ___ _____________
______   ________________________________________________ __  ___ _____________
.
.
.
______   ________________________________________________ __  ___ _____________
======   ===================== B O T T O M ============== ==  === =============
```

A series of literals and field names specifies a heading or footing on the report Heading fill-in. Although you enter the pieces of the heading or footing vertically on the fill-in, they are arranged horizontally across the top of each page of the report. A heading wider than the report width wraps around to the next line.

To define a heading or footing, enter the information in the following fields on the Heading fill-in:

Command

An area where the user can enter CA Ideal editing line commands. You can position the command area on the right or left of the screen using the SET EDIT MARGIN command. For more information, see the *Command Reference Guide.*

Field Name, Literal, Function, or Arithmetic Expression

The group-identifier, field name, literal, function, or arithmetic expression used as a part of the heading or footing. This area must be left blank if L*nn* is specified in the TAB column.

To modify the specification, the ERASE EOF key or the DELETE key can delete this value. When a value in this field is deleted, CA Ideal deletes corresponding options specified in that line.

Group Identifiers

Identify all headings and footings with a group label, which indicates the function of the heading or footing in the report. Valid group identifiers are:

- **<<RH>>** Specifies that the following lines define a report heading that prints on a separate page at the beginning of the report. You can include the report date in the heading by using report functions.
- **<<RF>>** Specifies that the following lines define a report footing that prints at the end of the report. You can use the TAB field to define the footing lines to print on the same page or a separate page. You can include summary information in the report footing by using report functions.
- **<<PH>>** Specifies that the following lines define a page heading that prints at the top of each page of the report. If no group identifiers are specified, any lines defined on the Heading fill-in are assumed to be part of a page heading. You can include the report date and page in the heading by using report functions.

  If multiple page headings <<PH>> are specified, the first page heading encountered is the one that appears on the report. If both a page title (on the Parameter fill-in) and a page heading (on the Heading fill-in) are specified, the page title from the Parameter fill-in appears on the report.

  When a TAB of L*nn* is specified for the first line of a page heading, the heading text is moved *nn* lines down from where the heading would have started. The TAB does not affect the automatic date and page number. They continue to print on the original first line.
- **<<PF>>** Specifies that the following lines define a page footing that prints at the bottom of each page of the report. You can include summary information in the page footing by using report functions.
- **<<BH>>** Specifies that the following lines define a control break heading to print whenever the value of a control break field changes. See the section titled *Detail Definition Fill-in* later in this chapter for the definition of a control break field. The heading text specified here overrides the default control break heading for the designated level.

  The control break heading identifier is specified with a level number to identify the control break level for which the heading is printed. The format for the control break group identifier is <<BH>> [LEVEL] *n*. The keyword LEVEL is optional. If no level is specified, the heading applies to all control breaks defined. You can specify more than one level, as in:

  ```
  <<BH>> LEVEL 1 2 3
  ```

  You can suppress an individual control break heading for any level by creating a <<BH>> group for that level with no lines following.

  You can include the report date and page in the heading by using report functions.

- **<<BF>>** Specifies that the following lines define a control break footing printed whenever the value of a control break field changes. See the section titled *Detail Definition Fill-in* later in this chapter for the definition of a control break field. The footing text specified here overrides the default control break footing for the designated level.

  The control break footing identifier is specified with a level number to identify the control break level for which the footing is printed. The format for the control break group identifier is <<BF>> [LEVEL] *n*. The keyword LEVEL is optional. If no level is specified, the footing applies to all control breaks defined. You can specify more than one level, as in:

  ```
  <<BF>> LEVEL 1 2 3
  ```

  You can suppress an individual control break footing for any level by creating a <<BF>> group for that level with no lines following. Specify LEVEL 0 as the <<BF>> group level to print grand totals at the end of the report.

  You can include summary information in the control break footing by using report functions. See the section titled *Functions* later in this appendix. If a <<BF>> group is defined, any of these functions specified on the Detail fill-in is ignored. The function must be defined in the <<BF>> group.

  Note: A double-space is automatically generated before a control break footing is produced.

- **<<EMPTY>>** Specifies that the following text prints when no PRODUCE statements execute for the report. The text to print must be composed of literals. No page or report headings print. If an <<EMPTY>> group is not specified and no PRODUCE statements execute, there is no output.

- **<<ABORT>>** Specifies that the following text prints when the value of the return code is equal to or greater than the value specified in the group identifier, which is specified as <<ABORT>> RC *x*. If the <<ABORT>> group is printed, the final control break, page, and report footings do not print. If an <<ABORT>> group is not specified, a standard message prints.

  The lines defining the output in an <<ABORT>> group are printed automatically when the program terminates abnormally. To force the <<ABORT>> group to print when releasing a report dynamically, use the statement RELEASE REPORT WITH ABORT.

  You can use the <<ABORT>> group with user-defined error procedures. Although you can code only one <<ABORT>> group, you can specify one or more working data variables to define the output, provided that the variables are current for the last PRODUCE REPORT statement executed. These variables can then be assigned in the error procedure based on the value of the return code ($RC). Variables set after the last PRODUCE REPORT statement was executed are not available for the <<ABORT>> group.

  The group identifier must appear on a line by itself. It is followed by lines that define the specified heading or footing. If only one heading is specified, no group identifier is required and the heading is assumed to be the page heading.

Field names

Field names must be the names of fields in dataviews, panels, working data, or parameters. You can continue field names for a total of up to three lines by terminating the name with a semicolon. For more information, see the section *Considerations for Continuation* later in this chapter.

Literals

You must enclose literals in single quotes. You can specify literals on consecutive lines with a TAB specification of 0 or 1. They are concatenated in the page heading. If spaces are required between the two literals, you can use the TAB specification to provide them or include them in the quotes. You cannot continue a heading literal with a semicolon.

Functions

Report functions entered in a heading or footing return a value when the heading or footing is produced. The following report functions are available:

- **$RPT-DATE** Specifies the current date as part of the heading. The date value is obtained from an ASSIGN REPORT statement, if one is specified. Otherwise, the current date is used. You can use this function for report, page, and control break headings.
- **$RPT-PAGE** Specifies the current page number as part of the heading. The page number value is obtained from an ASSIGN REPORT statement, if one is specified. Otherwise, a value of 1 is used. You can use this function for report, page, and control break headings.
- **$TOT(field)** Computes the total value for all occurrences of the specified field in the range covered by the footing. If $TOT is specified in the page footing, the value is the total value of all occurrences on the page. If it is specified in the control break footing, the value is computed based on all occurrences in the control break group. You cannot use this function in the report footing.

The $TOT function provides the same value as the TOT function on the Detail fill-in.

- **$AVG(field)** Computes the average value of all occurrences of the specified field in the range covered by the footing. You can use this function in all footings. The $AVG function provides the same value as the AVG function on the Detail fill-in.
- **$MIN(field)** Determines the minimum value of all occurrences of the specified field in the range covered by the footing. You can use this function in all footings. It provides the same value as the MIN function on the Detail fill-in.
- **$MAX(field)** Determines the maximum value of all occurrences of the specified field in the range covered by the footing. You can use this function in all footings. It provides the same value as the MAX function on the Detail fill-in.

- **$OCC[(field)]** Counts the non-null occurrences of a primary or secondary field if a field name is specified. If no field name is specified, $OCC counts the primary produce statements in the range covered by the footing. You can use this function in all footings. When specified with a field name, it provides the same value as the OCC function on the Detail fill-in.
- **$CTOT(field)** Accumulates the total value for all occurrences of the specified numeric field in the range of the report. You can use this function in all footings, but it always shows the total for the entire report.

The value entered for *field* in any of the above functions must be a unique name or label on a line in the Detail fill-in. If the field name is a qualified name (for example, EMPLOYEE.SALARY) in the Detail definition, then it must also be qualified when used with these functions in either the Heading or Footing definition. You can continue the function specification on a total of up to three lines by ending each line except the last with a semicolon.

Arithmetic Expressions

You can specify an arithmetic expression with or without a label. However, you cannot use the value computed for the expression as the operand of a report function if the expression is not labeled. You can continue an expression for a total of up to three lines by ending the Field Name entry in each line except the last with a semicolon.

Considerations for Continuation:

- Anything entered to the right of the semicolon, but in this column (Field Name, Literal, Function or Arithmetic Expression), is ignored.
- Leading blanks are ignored on subsequent lines.
- Continuing a function follows the same rules as in Program Definition Language.
- You can use a semicolon to indicate that the entry's edit pattern is continued.
- You must enter any other specifications for a multiple-line entry on the first line in the appropriate column.

Column

Specifies the width of the column and its placement on the report line.

Width

A number from 0 through 99 that specifies the width of the column. An entry here overrides the default column width. If no value is specified here (the default), the width is equal to the longest of the following:

- The width of the source field
- The width of the column head
- If a summary total is specified, then a width of four characters longer than the source field width is used
- The width that results from the edit pattern

If the field value is longer than the space allotted, it is truncated. Alphanumeric fields are truncated from the right. Numeric fields are truncated from the left.

If an alphanumeric function is specified, a WIDTH specification is required. See the PDL chapter in the *Programming Reference Guide* for a definition of terms.

Tab

Specifies the column and line where this field in the heading or footing starts. If omitted, the Spacing Between Columns option in the report Parameters fill-in determines the spacing. You can use the following entries:

- **+nn** Specifies relative spacing, where nn is the number of columns to skip from the current position.
- **nnn** Specifies absolute spacing, where nnn is a specific column number.
- **MAX** Specifies absolute spacing, where the field is positioned all the way to the right of the heading line (right justified).
- **Lnn** Specifies vertical spacing, where nn refers to the number of vertical spacing units. The Spacing Between Lines option in the Parameter fill-in determines the number of lines in a vertical spacing unit. The default value for *nn* is 1.

L*nn* must be a separate tab entry with all other columns blank. In the following example, it is necessary to leave the second line in the field name column blank because the TAB value is L05.

```
FIELD NAME   ...   TAB
X                   50
                   L05
Y                   50
```

Leaving the field name blank when L*nn* is specified in the TAB column is useful for skipping extra lines without producing a field on the report in that position.

Edit Pattern

Specifies an override to the default edit pattern for the entry. Refer to the edit pattern description in the section titled *Detail Definition Fill-in* later in this chapter. You can continue the edit pattern for a total of up to three lines by ending the field name column with a semicolon. You can also specify date patterns here; however, $RPT-DATE uses the date format specified in the Parameters fill-in. Be sure that the edit pattern is large enough to accommodate the largest possible value and provide a sign if negative values are possible.

Detail Definition Fillin

The report Detail definition fill-in specifies the fields to appear in each detail line of the body of the report and to specify control breaks, summary functions, and so on, for these fields. The report Detail fill-in can contain one primary group and multiple secondary groups. These primary and secondary groups allow for several different levels and sublevels of data to produce in a single report.

Report Groups

A report-group is a uniquely identified set of lines on a detail panel. A PRODUCE statement activates a report-group. Each group is identified by a group-name in angled braces appearing on a line by itself in the Field Name column.

```
<<group-name>>  [GROUP]
```

The lines following the identifier define what is contained in that group. A report-group name must be between 3 and 15 characters in length.

PrimaryGroup

The primary group is the first group defined on the report Detail fill-in. If a group name is specified for the primary group, it must be placed on the first line of the Detail fill-in. If a group name is not specified, the first group of detail lines is assumed to be the primary group. Control breaks and report functions are allowed on a primary group. You can use default column headings or define user-specified column headings on the Column Headings fill-in.

SecondaryGroup

All other groups defined on the Detail fill-in are secondary-groups. Report functions are allowed on secondary-groups, but control breaks are not allowed. You cannot define column headings in the same manner as headings for primary columns. See the section titled *Defining Column Headings* in Chapter 3 for an example of secondary group headings.

Displaying the Detail FillIn

To display the report Detail fill-in, enter the command DETAIL on the command line or press the F6 key. If you are editing an existing report definition, the EDIT command places you directly in the Detail fill-in.

The Detail fill-in is shown below.

```
=>
 -------------------------------------------------------------------------------
--
 IDEAL   : RPT DETAIL DEFN.     RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

        Field Name, Literal,      Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression     V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
 ------  -------------------------  - -   - - -  - - - -  - -- --- ------------
 ======  =========== T O P ======  = =   = = =  = = = =  = == === ============
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ______  _________________________  _ _   _ _ _  _ _ _ _  _ __ ___ ____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === ============
```

Command

Enter CA Ideal editing line commands here. Refer to the editing section in the *Command Reference Guide* for information about using editing line commands.

Field name, literal, function, or arithmetic expression

Specifies the report group name, field name, literal, function, or arithmetic expression to print on the detail line. This area must be left blank if you specify L*nn* or P in the TAB column.

If a field name, function, or arithmetic expression does not fit on one line, you can continue it on a subsequent line by terminating the entry text with a semicolon immediately following the field name and continuing on the next line. Leading blanks are ignored on subsequent lines. You cannot continue a literal with a semicolon. You can specify literals on consecutive lines with a TAB specification of +0. They are concatenated in the detail. To leave a space between the two pieces of the literal, you can specify a TAB value of +1 or include the space as part of one of the pieces of the literal.

For information in the following fields of the Detail fill-in: You can also use a semicolon here to continue the entry's edit pattern. Any other specifications for a multiple-line entry must be indicated on the first line of the appropriate column.

To modify the specification, use the ERASE EOF key or the DELETE key to delete a field name that is no longer wanted. When you delete a field name, the corresponding options specified on that line are also deleted.

Group Name

Specifies a label for the detail group. A group name is specified in the following format:

```
<<label>> [GROUP]
```

If specified, you must enter a group name must be entered on a line by itself before the first line of the detail definition it identifies.

Field Name

Field names must be the names of fields in dataviews, panels, working data, or parameter data. For field names, the following conditions apply:

- For alphanumeric and variable fields, the maximum field length is used unless a column width is specified. You can specify an edit pattern.
- For numeric fields, you must specify an edit pattern.
- For date fields, you can specify an edit pattern. If one is not specified, the date format is the default defined by SET COMMAND DATEFOR.
- For dataview fields, the dataview name must prefix the field name.

Fields specified with null values appear as question marks padded with blanks to the total field length. If alphanumeric, the field is left justified, followed by blanks. If numeric, the field is right justified, preceded by blanks.

Function

You can specify any CA Ideal PDL function. You can also specify the report function $OCC as the field name to print the count of primary records.

Arithmetic Expression

An expression that derives a result from an arithmetic operation using field names, literals, or functions as operands. Arithmetic expressions can be named or unnamed. You can use the result of a named expression in another expression.

Sort

Specifies a sorting order for detail lines in the report different from the order in which lines of the report were produced. The sorting order is specified using two values, the sort level of the field and the direction of the sort.

For reports with no sorting specified, the order of the detail lines in the report is the order in which the lines were selected or produced. See the ORDERED BY clause on the FOR EACH statement in the *Programming Reference Guid*e for information on how to sequence selected records. The FOR EACH statement in the application procedure selects data and usually accompanies the production of reports.

A primary-group and secondary-groups always appear in the order they were produced. For reports with a sorting order on a primary group, the primary group and its associated secondary groups remains together.

- **LVL (level)** Specifies the order in which fields are sorted. If omitted, this field is not sorted.
    - Specifies this field as the first or major sort field.
    - **2-9** Specifies this field as a second or a minor sort field. You can indicate up to nine levels of sort fields. You can assign these numbers to fields in any order, but they must form an unbroken sequence. (For example, 1,3,2 is acceptable, but 1,3,4 is not.)

Note: Fields containing null values are sorted after any other numeric or alphanumeric values.

A/D (Ascending/ Descending)

Specifies whether the sort is in ascending or descending order.

- **A** Sort in ascending order (default).
- **D** Sort in descending order.
- **+(plus)** Sort in ascending order, based on the alternate collating sequence, if one has been defined through the site's sort package.
- **-(minus)** Sort in descending order, based on the alternate collating sequence, if one has been defined through the site's sort package.

Break

Specifies the format of a control break. Control breaks are not allowed on secondary groups.

- **LVL (level)** Establishes control fields and their relative levels. If a value is not entered here, the field is not used as a control field.
  - **1-9** Specifies the field as a control break field. The control break level is assigned in decreasing order of importance as the number increases. For example, LVL 1 is the first or major control break field; LVL 2 is the next; and so on. You can assign control break levels to fields in any order, but they must form an unbroken sequence without duplicate numbers; for example: 1,3,2 is acceptable, but 1,3,4 or 1,2,2 is not.
- **SKP (skip)** Indicates the spacing from the previous control footing to the current control heading when a control break occurs. If this field is left blank, the spacing is the same as the Spacing Between Lines option specified in the Parameter fill-in.
  - **P** Positions the report to the top of a new page.
  - **R** Resets the page number to 1 and positions the report to the top of a new page.
  - **1-9** Specifies the number of lines to skip.

You can override this field with the TAB specification for the control break heading <<BH>> in the Heading fill-in.

- **IND (indication)** Specifies whether the control field value prints to indicate the beginning of a new group, for all detail lines, or not at all. If nothing is specified here, the control field prints only when the value changes.
- **G** Control field prints in detail lines when the value changes.
- **R** Control field always prints (repeats) in all detail lines.
- **N** None, control field does not print in detail lines.

Function

Specifies summary operations to perform against this field. The resulting value prints in all user-defined break level footings and in standard control breaks: standard break levels, page footings, and report footings. If the function column is left blank, the function is not performed and no value prints for that field.

For each of the following functions, you can specify A for annotated or S for summary:

- **A** Prints an annotated value on a separate line. The annotation consists of the column heading and the field value. If the column heading normally prints on more than one line, the sections of the heading are concatenated to print on one line.
- **S** Prints all values on a single summary line beneath the corresponding columns.

  Note: When any of these functions are requested, the default width of the associated detail field is computed as the implied width of the detail field plus 4. You can override this default explicitly in the Width field.
- **TOT (total)** Specifies whether a total is computed and printed.

A question mark in the leftmost position of a summary total indicates either that the user-defined edit pattern is too small or, when no edit pattern is specified, that the TOTAL requires more than four extra positions.

- **MAX (maximum)** Specifies whether the maximum value that occurs for the field in the control break group is computed and printed.
- **MIN (minimum)** Specifies whether the minimum value that occurs for the field in the control break group is computed and printed.
- **AVG (average)** Specifies whether the average of all values of a field in the control break group are computed and printed. Nullable fields cannot specify the AVG option.

Column

The options under this heading specify whether column headings are printed, define the width of the report column, and specify the placement of the column on the detail line.

- **HDG (heading)** Specifies what is used as the column heading for fields in the primary group.
  - **N** No heading is printed.
  - **U** The user-defined heading specified on the Column Heading fill-in is used. The U appears automatically when a heading is defined on the Column Headings fill-in.

If you leave this field blank, the column heading specified in the dictionary facility is used if available. If no column heading is specified in the dictionary, the field name is used. When a literal value is defined on a detail line, the column heading for the literal value *must* be user-defined. In this case, you cannot leave the HDG option blank.

Note: The column heading derived from the cataloged dataview contains a maximum of 20 characters, even though the cataloged dataview heading fields can have up to 36 characters and the field names can be up to 32 characters. You can define longer headings on the Column Heading fill-in.

- **WIDTH** A value of 0 to 99 that specifies the width of the column. A width of zero suppresses printing of the field on the detail line but allows MIN, MAX, AVG, and TOT to print as annotations only. An entry here overrides the default column width. If the field value is larger than the space allotted, it is truncated. Truncation of alphanumeric fields is on the right. Truncation of numeric fields is on the left.

If this field is alphanumeric and blank, the width is equal to the largest of the following:

- The width of the source field.
- The width of the column heading.

If this field is numeric and blank, the width is equal to the largest of the following:

- The width of the column heading.
- The width that results from the edit pattern.
- A width of four characters larger than the edit pattern width if a summary total is specified.

If an alphanumeric function such as $SUBSTR is specified, a WIDTH specification is required. When using an alphanumeric function, the width must be a positive number greater than zero. The column width is required to define the length for a function's result, otherwise a compile error occurs.

TAB

Specifies the starting point for the field in the detail line. If this field is blank, the value is taken from the Spacing Between Columns option in the Parameter fill-in.

- **+nn** Specifies relative spacing, where nn is the number of columns to skip from the current position.
- **nnn** Specifies absolute spacing, where nnn is the specific column number. When absolute spacing is used and nnn is to the left of the current position, the line is continued on the next line at the absolute position.
- **MAX** Specifies absolute spacing, with the field right justified on the output line.
- **Lnn** Specifies vertical spacing, where nn refers to the number of vertical spacing units. The default value for nn is 01. The number of lines in each vertical spacing unit is taken from the Spacing Between Lines option in the Parameter fill-in.
- **P** Specifies vertical spacing, positioning at the top of the next page.

  L*nn* and P must be separate entries with blank field names. In the following example, it is necessary to leave the second and fourth field name columns blank because of the value in the TAB column.

  ```
  FIELD NAME   ...   TAB
  X                   50
                     L05
  Y                   50
                       P
  ```

  Leaving the field name blank when you specify L*nn* in the TAB column is useful for skipping extra lines without producing a field on the report in that position.

Edit Pattern

Specifies the format in which the value of the field prints. An edit pattern is required when a numeric function or an arithmetic expression is specified and is optional for other data types.

For alphanumeric and numeric data items, the pattern entry is a sequence of any of the edit pattern symbols. In addition, an L in the first character of the edit pattern left justifies the result.

For date type data items, the entry is a sequence of characters representing the date pattern as defined in the $DATE function section in the *Programming Reference Guide*.

The edit pattern can take up to three lines of the Edit Pattern Field. However, the maximum length of an edit pattern is 30 characters. A semicolon in the name column is used as a continuation symbol. The semicolon must appear immediately following the field name to continue. Spaces found anywhere in the edit pattern are not suppressed.

Note: The numeric date functions such as $TODAY and $INTERNAL-DATE require an edit pattern, not a date pattern. The report functions $RPT-DATE and $RPT-PAGE should not use a date or an edit pattern.

If this field is blank, the edit pattern is determined as follows:

- If omitted and if an edit pattern for the field is specified in the dictionary facility, that edit pattern is used.
- If an edit pattern is not found in the dictionary facility for this field, then the following rule is used:
  - If the field is alphanumeric, left justify the value and blank fill.
  - If the field is numeric, right justify the value and suppress leading zeros. If decimal places are present, replace with a period. If the value is negative, put a minus sign on the right-hand side.
  - If the field is a date field, use the default pattern for the site or the pattern set by a SET COMMAND DATEFOR command. For more information for an explanation of this command, see the *Command Reference Guide*.

Category and Meaning		Data in Source	Edit Pattern	Result
Pattern characters for alphanumeric data				
x	Alphanumeric character	STATE	X(5)	STAT
	Any other character represents itself	AB1234	XXX-XXX	AB1-234

Category and Meaning		Data in Source	Edit Pattern	Result
Pattern characters for numeric data				
L	Left-justify before output	123	LZ(3),ZZ9**	123
9	Unsuppressed numeric digit	123	999	123
Z	Zero suppression	v12*	ZZZ.99	.12
*	Asterisk replacement	lv23	***9.99	***1.23
,	Comma	002234v56	ZZZ,ZZZ.99	2,234.56
/	Slash	123083	99/99/99	12/30/83
B	Blank space	123083	99B99B99	12 30 83
0	Zero	123	99900	12300
.	Decimal point	030v99	ZZZ.99	30.99
-	Minus sign, fixed right	-23v45	9(2).99-	23.45-
		23v45	9(2).99-	23.45
-	Minus sign, fixed left	-23v45	-9(21).99	-23.45
-	Minus sign, floating	-v23	--.99	-.23
+	Plus sign, fixed right	67v89	9(2).99+	67.89+
+	Plus sign, fixed left	67v89	+9(2).99	+67.89
+	Plus sign, floating	00v67	++.99	+.67
CR	Credit symbol, right	-25v00	9(2).99CR	25.00CR
		25v00	9(2).99CR	25.00
DB	Debit symbol, right	-13v00	9(2).99DB	13.00DB
		13v00	9(2).99DB	13.00
$	Dollar sign, fixed	004v00	$Z9.99	$4.00
$	Dollar sign, floating	001v23	$$$.99<9,999. 99>	$1.23
<>	Encloses negative numbers in parentheses, fixed	-23v45		(0,023.45)
<>	Encloses negative numbers in parentheses, floating	-23v45	$<,<<<.99>	$ (23.45)

* Lowercase v represents the position of an assumed decimal point.

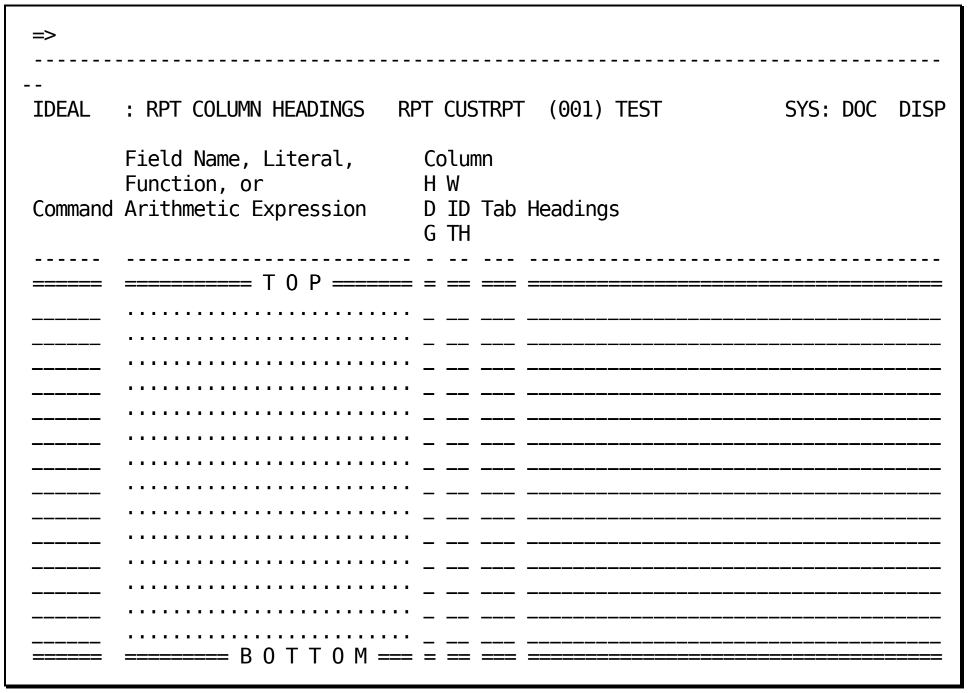
** You can condense edit patterns by using multipliers. For example, you can specify the expanded pattern ZZZZZZZZZ.99999 in the edit pattern as Z(9).9(5).

For a complete explanation of the rules summarized in this table, see the *CODASYL COBOL Journal of Development*.

Column Headings Definition Fillin

The report Column Headings definition fill-in overrides a column heading defined in the dictionary facility for a primary-group field. Use of this fill-in is optional.

To display the Column Headings fill-in, enter the command COLUMN on the command line or press the F6 key when the Detail fill-in is displayed.



```
 =>
 -------------------------------------------------------------------------------
--
 IDEAL   : RPT COLUMN HEADINGS   RPT CUSTRPT  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Column
         Function, or              H W
 Command Arithmetic Expression     D ID Tab Headings
                                   G TH
 ------  ------------------------- - -- --- ------------------------------------
 ======  =========== T O P ======= = == === ====================================
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ______  ......................... _ __ ___ ____________________________________
 ======  ========= B O T T O M === = == === ====================================
```

To define a column heading to use in place of the default column heading, enter the necessary information in the following fields:

Command

An area where the user can enter CA Ideal editing line commands. You can position the command area on the right or left of the screen using the SET EDIT MARGIN command. For more information, see the *Command Reference Guide*.

Field name, literal, function, or arithmetic expression

Contains the group name, field name, literal, function, or arithmetic expression defined on the Detail fill-in. You can modify this field or, if not entered in the Detail fill-in, enter it here. For more information, see the section *Detail Definition Fill-in*.

Column

Repeats the column options specified on the Detail fill-in. You can make modifications to this entry on either fill-in. For more information, see the section titled *Report Detail Definition Fill-in* earlier in this chapter.

Headings

Specifies the text to use as the column heading. You can extend this specification over three lines of the fill-in at most. A semicolon in the Field Name column is the continuation symbol.

You can print the column heading on one to five lines. If the first character of the heading literal is not a blank, alphabetic, or numeric, that character is considered to be a delimiter. For example, in the specification /TYPES/OF/EMPLOYEES, the first character is used as the delimiter and subsequent slash characters are treated as additional delimiters indicating the beginnings of new lines, as in:

```
TYPES
OF
EMPLOYEES
```

If you want the column heading on the report to appear as

```
*** TYPES OF EMPLOYEES ***
```

you must specify another non-alphabetic or numeric character before the asterisks, such as /*** TYPES OF EMPLOYEES ***. Otherwise, the first asterisk is taken as a delimiter.

Appendix B: Sample Reports

The first two sample reports in this Appendix are produced using the EMPLOYEE dataview. The third sample report, in the section Mailing Labels, uses the CUSTOMER dataview.

```
=>
 -----------------------------------------------------------------------------
--
 IDEAL   : DISPLAY DATAVIEW      DVW EMPLOYEE  (001) TEST      SYS: DOC  FILL-IN

 Commnd Seq Level Field name           T I Ch/Dg Occur Value/Redef/Dep on
 ====== === ===== ====== T O P ======= = = ===== ===== =========================
 000100 CATALOGED 02/23/95 05:17       SEQUENTIAL      EMPLOYEE UPD=YES
 000200   1 1     EMPLOYEE
 000300   2  2    NUMBER               U Z     5
 000400   3  2    NAME                 X      24
 000500   4  2    STREET-ADDRESS       X      24
 000600   5  2    CITY-ADDRESS         X      15
 000700   6  2    STATE-ADDRESS        X       2
 000800   7  2    ZIP-CODE-LOC         X       5
 000900   8  2    SOCIAL-SECURITY      N P     9
 ====== === ===== === B O T T O M ==== = = ===== ===== =========================
```

Sample Report 1

The following is sample report #1:

```
              ❶ **EMPLOYEE REPORT FOR    CA , IL    **                ❷ PAGE 1
❸ NAME                          STREET                         SALARY
❹  _______________________   _______________________ ❿_____________
    FONTANA WALLACE            3821 GREENBRIAR CR            14770.00
    HAYWOOD VANCE              107 FANNIN AVE                14090.00
    MORAN SIMON                1353 OAKBLUFF                 12550.00
    PRICE JOEL                 6321 ORIOLE DR                14320.00
    WILKES EARL                9990 MONROE AVE               15760.00
❺       CITY-ADDRESS LOS ANGELES
❻ TOTAL                                                    71490.00
        COUNT         5
    NOLAND TURNER              3570 MAVERICK DR              12940.00
    PATMAN ROBERT              3918 HAWICK AVE               14850.00
    PELTON LEWIS               5516 OUTRIDER CT              13810.00
    WREN PATRICIA              6127 GASTON AVE                    .00
❺       CITY-ADDRESS SAN FRANCISCO
❻  TOTAL                                                    41600.00
        COUNT         4
❼     STATE-ADDRESS CA
❽  TOTAL                                                   113090.00
         COUNT       9


                **EMPLOYEE REPORT FOR    CA , IL    **                ❾ PAGE 2
    NAME                          STREET                         SALARY

    _______________________   _______________________   _____________
    GOODWIN PATRICK            4308 ASTOR DR                 12180.00
    LINDSTROM ROGER            2064 FAROLA LA                12960.00
    WASHBURN STELLA            1711 HERALD ST                13950.00
         CITY-ADDRESS CHICAGO
    TOTAL                                                    39090.00
        COUNT         3
       STATE-ADDRESS IL
    TOTAL                                                    39090.00
         COUNT       3
    TOTAL                                                   152180.00
       COUNT    12
????????????????????①????????????????????
```

❶	Page heading
❷	Page number, top right
❸	Column heading
❹	Column heading highlight, underscores
❺	Control break footing for control break level 2
❻	Annotated summary line for control break level 2
❼	Control break footing for control break level 1
❽	Annotated summary line for control break level 1
❾	Page break
❿	Spacing between columns
①	Report width

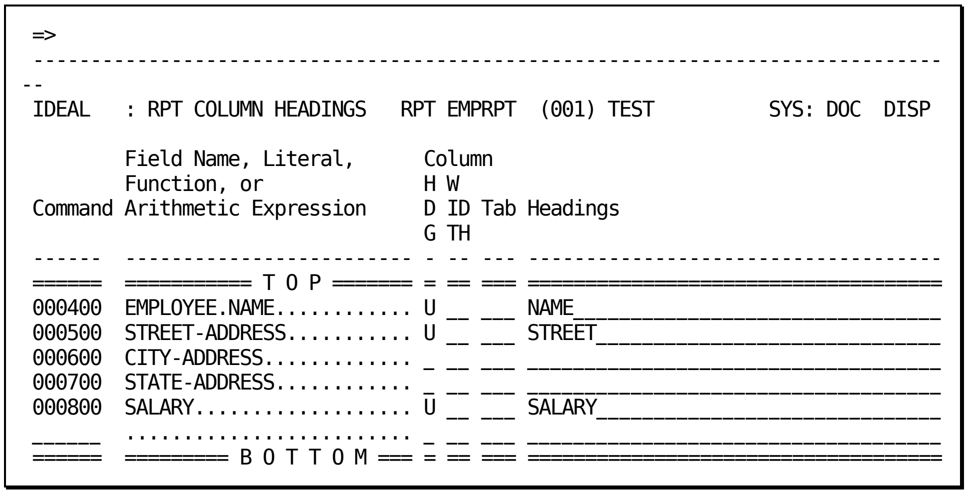
The following screens show the fill-ins completed for sample report #1. This screen shows the completed report parameters fill-in for sample report #1.

```
 =>
-------------------------------- Partially shown
 IDEAL   : RPT PARAMETERS       RPT EMPRPT  (001) TEST           SYS: DOC  DISP

 Lines per page on printout           060  (0 thru 250)
 Report width                         080  (40 thru 230)
Spacing between lines                1    (1 thru 3)
Spacing between columns              02   (0 thru 66 OR A=Automatic)
Spacing after page and column hdgs   1    (0 thru 9)
Summaries only                       N    (Y=Yes,N=No)
 Column headings desired              Y    (Y=Yes,N=No)
 Column headings indication           U    (U=Underscore,N=None,D=Dashes)
Control break heading                N    (Y=Yes,N=No)
 Control break footing                Y    (Y=Yes,N=No)
 Automatic page footing summaries     N    (Y=Yes,N=No)
 Group continuation at top of page    Y    (Y=Yes,N=No)
 Annotated count in break footings    Y    (Y=Yes,N=No)
 Report final summary title           N    (Y=Yes,N=No)
   Spacing before summary             2    (1 thru 9 = Lines,P=New Page)
   Title     __________________________________________

 Date
   Position           NO   (NO=None, BR=Bot.Right, BL=Bot.Left, BC=Bot.Ctr.,
                            TR=Top Right, TL=Top Left, TC=Top Center)
   Format             MM/DD/YY

 Page Numbers
   Position           TR
   Format             P    (D=Digits Only, H=With Hyphens, P= Page      nnn)

 Page Heading
   Heading            __________________________________________
   Position           L    (C=Center, L=Left Justify, R=Right Justify)
```

This screen shows the completed report detail fill-in for sample report #1.

```
 =>
 -------------------------------------------------------------------------------
--
 IDEAL   : RPT DETAIL DEFN.     RPT EMPRPT  (001) TEST           SYS: DOC  DISP

        Field Name, Literal,      Sort   Break  Function Column
        Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression     V /   V K N  O A I V  D ID Tab Edit Pattern
                                   L D   L P D  T X N G  G TH
 ------  -------------------------  - -   - - -  - - - -  - -- --- -------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === =============
 000400  EMPLOYEE.NAME              _ _   _ _ _  _ _ _ _  U __ ___ _____________
 000500  STREET-ADDRESS             _ _   _ _ _  _ _ _ _  U __ ___ _____________
 000600  CITY-ADDRESS               _ _   2 2 N  _ _ _ _  _ __ ___ _____________
 000700  STATE-ADDRESS              _ _   1 P N  _ _ _ _  _ __ ___ _____________
 000800  SALARY                     _ _   _ _ _  S _ _ _  U __ ___ _____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

This screen shows the completed column headings fill-in for sample report #1.



```
 =>
 -----------------------------------------------------------------------------
--
 IDEAL   : RPT COLUMN HEADINGS   RPT EMPRPT  (001) TEST         SYS: DOC  DISP

         Field Name, Literal,      Column
         Function, or              H W
 Command Arithmetic Expression     D ID Tab Headings
                                   G TH
 ------  ------------------------- - -- --- ------------------------------------
 ======  =========== T O P ======= = == === ====================================
 000400  EMPLOYEE.NAME............ U __ ___ NAME________________________________
 000500  STREET-ADDRESS........... U __ ___ STREET______________________________
 000600  CITY-ADDRESS............. _ __ ___ ____________________________________
 000700  STATE-ADDRESS............ _ __ ___ ____________________________________
 000800  SALARY................... U __ ___ SALARY______________________________
 ______  ......................... _ __ ___ ____________________________________
 ======  ========= B O T T O M === = == === ====================================
```

Sample Report 2

The following shows sample report #2.

```
                             ❶ -1-                      ❷ 01/18/95
                        **EMPLOYEE REPORT FOR  CA, IL  **
      NAME                STREET              CITY          STATE         SALARY
❸
      FONTANA WALLACE     3821 GREENBRIAR CR ❹ LOS ANGELES   CA         14770.00
      HAYWOOD VANCE       107 FANNIN AVE                                 14090.00
      MORAN SIMON         1353 OAKBLUFF                                  12550.00
❻     PRICE JOEL          6321 ORIOLE DR                                 14320.00
      WILKES EARL         9990 MONROE AVE                                15760.00
    ❺ CITY LOS ANGELES
      TOTAL                                                       ❻     71490.00
      NOLAND TURNER       3570 MAVERICK DR     SAN FRANCISCO             12940.00
      PATMAN ROBERT       3918 HAWICK AVE                                14850.00
      PELTON LEWIS        5516 OUTRIDER CT                               13810.00
      WREN PATRICIA       6127 GASTON AVE                                     .00
    ❺ CITY SAN FRANCISCO
❼     TOTAL                                                       ❻      41600.00
       STATE CA
❽ TOTAL                                                                 113090.00

                                    -2-                                  01/18/95
                        **EMPLOYEE REPORT FOR  CA, IL  **
      NAME                STREET              CITY           STATE        SALARY
      GOODWIN PATRICK     4308 ASTOR DR        CHICAGO     ❾ IL         12180.00
      JARRETT LOUIS       10451 CHANNEL DR                               15500.00
      LINDSTROM ROGER     2064 FAROLA LA                                 12960.00
      SAUNDERS LOUIS      1117 WADE DR                                   14750.00
      WALKER DENNIS       637 WILLOW ST                                  14690.00
      WASHBURN STELLA     1711 HERALD ST                                 13950.00
           CITY CHICAGO
      TOTAL                                                              84030.00
❼        STATE IL
      TOTAL                                                              84030.00
❿      REGIONAL SUMMARY
① TOTAL                                                                 197120.00
```

❶ Page number, centered and hyphenated
❷ Date, top right
❸ Highlighting, none
❹ Control break level 2
❺ Control break footing for control break level 2
❻ Annotated summary line for control break level 2
❼ Control break footing for control break level 1
❽ Annotated summary line for control break level 1
❾ Control break level 1
❿ Final summary title
① Final total

The following screens show the fill-ins completed for sample report #2.

This screen shows the completed report parameters fill-in for sample report #2.

```
 =>
-------------------------------- Partially shown
 IDEAL   : RPT PARAMETERS        RPT EMPRPT2  (001) TEST        SYS: DOC  DISP

 Lines per page on printout           060  (0 thru 250)
 Report width                         080  (40 thru 230)
Spacing between lines                1    (1 thru 3)
Spacing between columns              02   (0 thru 66 OR A=Automatic)
Spacing after page and column hdgs   1    (0 thru 9)
Summaries only                       N    (Y=Yes,N=No)
 Column headings desired              Y    (Y=Yes,N=No)
 Column headings indication           N    (U=Underscore,N=None,D=Dashes)
Control break heading                N    (Y=Yes,N=No)
 Control break footing                Y    (Y=Yes,N=No)
 Automatic page footing summaries     N    (Y=Yes,N=No)
 Group continuation at top of page    Y    (Y=Yes,N=No)
 Annotated count in break footings    N    (Y=Yes,N=No)
 Report final summary title           N    (Y=Yes,N=No)
   Spacing before summary             2    (1 thru 9 = Lines,P=New Page)
   Title    'REGIONAL SUMMARY'

 Date
   Position           TR   (NO=None, BR=Bot.Right, BL=Bot.Left, BC=Bot.Ctr.,
                            TR=Top Right, TL=Top Left, TC=Top Center)
   Format             MM/DD/YY

 Page Numbers
   Position           TC
   Format             H    (D=Digits Only, H=With Hyphens, P= Page       nnn)

 Page Heading
   Heading            __________________________________________
   Position           C    (C=Center, L=Left Justify, R=Right Justify)
```

This screen shows the completed Headings Fill-in:

```
=>
 ------------------------------------------------------------------------------
--
 IDEAL   : RPT HEADING/FOOTING   RPT EMPRPT2  (001) TEST          SYS: DOC  DISP

         Field Name, Literal, Function,                    Column
 Command  or Arithmetic Expression                         Wid Tab Edit Pattern
 ------  ------------------------------------------------- --  --- ------------
 ======  ==================== T O P ===================== ==  === ============
 000201  '**EMPLOYEE REPORT FOR'                           __  ___ ____________
 000202  STATE1                                            __  ___ ____________
 000203  ','                                               __  00 ____________
 000204  STATE2                                            __  01 ____________
 000205  '**'                                              __  ___ ____________
 ======  ==================== B O T T O M ============= ==  === ============
```

This screen shows the report detail definition fill-in for sample report #2.

```
=>
 -------------------------------------------------------------------------------
--
 IDEAL   : RPT DETAIL DEFN.      RPT EMPRPT2  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
         Function, or               L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  -------------------------  - -   - - -  - - - -  - -- --- -------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === =============
 000400  EMPLOYEE.NAME              _ _   _ _ _  _ _ _ _  U 20 ___ _____________
 000500  STREET-ADDRESS             _ _   _ _ _  _ _ _ _  U 20 ___ _____________
 000600  CITY-ADDRESS               _ _   2 2 G  _ _ _ _  U 13 ___ _____________
 000700  STATE-ADDRESS              _ _   1 P G  _ _ _ _  U __ ___ _____________
 000800  SALARY                     _ _   _ _ _  S _ _ _  U 12 ___ _____________
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

Mailing Labels

You can use CA Ideal RDF to generate mailing labels. The following sample program demonstrates the basic steps necessary to produce labels on standard stock sheets. The sheets used for this program contain three labels across and eight labels down per sheet.

The label size is 4 x 1.5 inches. The data for one label occupies an area of 3 x .5 inches. A carriage control tape is created with a channel-one punch for each row of labels on the stock page.

The sample program prints mailing labels for every customer. Each address label is formatted as:

```
Employee Name
Street Address
City State Zip
```

Parameter Specifications

For RDF to print just the data without headings and footings, titles, and so forth, use the Parameter Fill-in screen. The following specifications are needed:

```
=>
 ------------------------------------------------------------- Partially shown
 IDEAL   : RPT PARAMETERS       RPT CUSTLBL  (001) TEST          SYS: DOC  DISP

 Lines per page on printout           006  (0 thru 250)
 Report width                         132  (40 thru 230)
Spacing between lines                 1    (1 thru 3)
Spacing between columns               00   (0 thru 66 OR A=Automatic)
Spacing after page and column hdgs    0    (0 thru 9)
Summaries only                        N    (Y=Yes,N=No)
 Column headings desired              N    (Y=Yes,N=No)
 Column headings indication           N    (U=Underscore,N=None,D=Dashes)
Control break heading                 N    (Y=Yes,N=No)
 Control break footing                N    (Y=Yes,N=No)
 Automatic page footing summaries     N    (Y=Yes,N=No)
 Group continuation at top of page    N    (Y=Yes,N=No)
 Annotated count in break footings    N    (Y=Yes,N=No)
 Report final summary title           N    (Y=Yes,N=No)
   Spacing before summary             P    (1 thru 9 = Lines,P=New Page)
   Title     __________________________________________
    .
    .
    .
```

Each page in the report is a single row of labels. The lines per page value is 6. Data prints on the second, third, and fourth lines to center it on the label. The width of the page is set to 132 characters to accommodate the sheet of labels and provide for three labels across. All lines are single spaced as specified by 1 in the spacing between lines field.

All other values are defined as N or 00, as appropriate, except for spacing before summary. Here P is specified. These values turn off the automatic facilities inherent in RDF that provide headings, labels, column positions, and so forth.

Using a Table Defined in Working Data

This program is special because it prints three rows from the customer table using a single PRODUCE statement. To do this, a table is defined in working data to contain the information for the three rows. This table is formatted so that it can be used in the Detail Definition of the report for the print-line layout. Each field in the table is defined as an alphanumeric value of 30 characters.

```
=>
 -------------------------------------------------------------------------------
--
 IDEAL   : WORKING DATA DEFN.    PGM LBLPGM  (001) TEST          SYS: DOC  DISP

 Command Level Field name          T I Ch/Dg Occur Value/Comments/Clauses
 ------  ----- ------------------- - - ----- ----- -----------------------------
 ======  ===== ====== T O P ====== = = ===== ===== =============================
 000100  1     LABEL-DATA                        3
 000200   2    L-NAME              X      30
 000300   2    L-ADD               X      30
 000400   2    L-CTYST             X      30
 ======  ===== === B O T T O M === = = ===== ===== =============================
```

Detail Definition Provides Report Definition

The specifications in the Detail Definition are the most important. The report is basically defined on only 12 lines on the Detail Definition Fill-in screen. This is possible by using vertical tabbing.

```
=>
 -------------------------------------------------------------------------------
--
 IDEAL   : RPT DETAIL DEFN.      RPT CUSTLBL  (001) TEST          SYS: DOC  DISP

         Field Name, Literal,      Sort   Break  Function Column
        Function, or                L A   L S I  T M M A  H W
 Command Arithmetic Expression      V /   V K N  O A I V  D ID Tab Edit Pattern
                                    L D   L P D  T X N G  G TH
 ------  -------------------------  - -   - - -  - - - -  - -- --- -------------
 ======  =========== T O P =======  = =   = = =  = = = =  = == === =============
 000100                                                        P
 000200                                                        L02
 000300  L-NAME(1)                                          30 011
 000400  L-NAME(2)                                          30 052
 000500  L-NAME(3)                                          30 093
 000600                                                        L01
 000700  L-ADD(1)                                           30 011
 000800  L-ADD(2)                                           30 052
 000900  L-ADD(3)                                           30 093
 001000                                                        L01
 001100  L-CTYST(1)                                         30 011
 001200  L-CTYST(2)                                         30 052
 001300  L-CTYST(3)                                         30 093
 ======  ========= B O T T O M ===  = =   = = =  = = = =  = == === =============
```

Note the values in the TAB field of the display. The vertical placement or line spacing is specified by the following:

- **L02-**Skip two lines. Labels print on the second physical line.
- **L01-**Skip to next line.
- **P-**Skip to top-of-form. Advances form to next set of physical labels.

The horizontal placement or column spacing positions the three sets of address information. That is, for example, all three customer names are defined on the same physical line but positioned at consecutive label fields. The first value defines the length of the field (all are 30 characters as is the data). The second value defines the position on the line. For example, a WIDTH and TAB specification of 30 and 11; positions a 30-character name starting in position 11. As you can see, the next name is positioned starting in column 52. The third starts in column 93. All fields with the same subscript are positioned to begin in the same column to obtain correct address format.

With these specifications, a single PRODUCE statement prints all three labels. The top-of-form specification with the carriage control tape positions the next row of labels and readies it for printing.

One PRODUCE, Multiple Rows

For one PRODUCE statement to print multiple rows, the program must set the associated variables before executing the PRODUCE statement. In this program, a table is defined in working data to contain the address information for three customers. The same names are used in the detail definition to format the labels. Since PRODUCE automatically moves the values in the same-name working data variables to the detail fields, the program must assign the values to the working data variables. The following segment assumes a variable was defined in working data named IDX to index through the repeating group.

```
SET IDX = 0
FOR EACH CUSTOMER
   ORDERED BY CUSTOMER.NAME
   ADD 1 TO IDX
   SET L-NAME(IDX) = CUSTOMER.CUSTNAM
   SET L-ADD(IDX) = CUSTOMER.ADDRESS
   SET L-CTYST(IDX) =
         $STRING(CUSTOMER.CITY, CUSTOMER,STATE,
                 CUSTOMER.ZIP)
   IF IDX = 3
      PRODUCE CUSTLBL
      SET IDX = 0
   ENDIF
ENDFOR
```

As long as complete sets of three are available, the PRODUCE statement executes in the FOR construct. The index, IDX, is reset to zero and the assignment of the label information continues. When there are no more rows, the FOR construct terminates. This could result in one or two unprinted labels. To handle this, a simple IF construct tests the value of IDX after the FOR statement. If it is not zero, then some labels remain to print. The statements in the IF construct blank out the extra labels (to avoid reprinting a label from the last set) and PRODUCE the last labels.

```
IF IDX NOT = 0
   SET IDX = IDX + 1
   LOOP VARYING IDX FROM IDX BY 1 THRU 3
       MOVE $SPACES TO L-NAME(IDX)
                       L-ADD(IDX)
                       L-CTYST(IDX)
   ENDLOOP
   PRODUCE CUSTLBLS
ENDIF
```

A subprocedure might be necessary to allow the computer operator to align the label sheet properly at the beginning of the run. You can use the following subprocedure before the FOR construct with the DO statement:

```
<<ALIGN>> PROCEDURE
   LOOP 5 TIMES
      LOOP VARYING IDX FROM 1 BY 1 THRU 3
         MOVE STAR-LIT TO L-NAME(IDX)
                          L-ADD(IDX)
                          L-CTYST(IDX)
      ENDLOOP
      PRODUCE CUSTLBLS
   ENDLOOP
ENDPROC
```

STAR-LIT is a literal value defined in working data. This value consists of a string of 30 asterisks that aligns the sheet.