

## Report Writer

---



---

Report Writer Overview.....	27-2
General Idea of the Report Writer.....	27-2
The Report Definition.....	27-2
HELP Section.....	27-3
VARIABLENAMES Section.....	27-4
VARIABLES DEFINITION Section.....	27-4
HEADER Section.....	27-5
DETAILLINE Section & DETAILLINE2 Section.....	27-5
SUBTOTALSLINE, SUBTOTALSLINE2 and GRANDTOTALSLINE Sections.....	27-6
SETUP Section.....	27-6
RECORDSELECTION Section.....	27-7
SORTDEFINITION Section.....	27-9
PRINT Section.....	27-10
PRINTEOF Section.....	27-11
Report Writer Selection.....	27-11
Report Description File.....	27-13
File Override.....	27-14
Report Title.....	27-14
Record Selection Output File.....	27-14
Sorted Record Output File.....	27-14
Screen Variables.....	27-15
Status/Error Lines.....	27-15
Report "Pause".....	27-15
Predefined Variables.....	27-16
&KEYNUM/&KEYSEGN Combinations by File Id (FID).....	27-19
Valid Arithmetic And Logical Connectors.....	27-21
Report Writer Menu.....	27-22
Summary Rankings.....	27-23
Include/HInclude Files.....	27-25

## Report Writer Overview

The Broker/Owner Orders Management System (BOOMS) Report Writer provides an environment for creating customized reports from BOOMS data (Orders, Lists, Clients, Accounts Receivable, Accounts Payable, General Ledger, etc). These reports can be used to supplement those produced by the built-in reporting facilities of BOOMS. Although many of the existing reports could instead be derived by using the Report Writer, the Report Writer should not be viewed as a replacement for these reports. Rather, it should be seen as a means of obtaining new information from your existing data. Lissan Computing Company is committed to providing an effective reporting environment for BOOMS. In addition, the users of the system are encouraged to share their Report Writer ideas with each other. LCC will be happy to coordinate any such effort and distribute new report formats with future updates to BOOMS. A supplemental document, "Summary of Report Writer Reports", is included with each new version of the system. Refer to that document for a current list of available reports.

## General Idea of the Report Writer

The Report Writer is designed as a reporting environment. Although this would seem apparent by its name, this point must be stressed in order to eliminate any confusion that may arise regarding what the Report Writer can do and what it cannot do. It is not designed to give a full set of programming features. These types of features are what have already been incorporated into BOOMS.

The Report Writer allows you to:

- Sort BOOMS data in a variety of ways
- Select records to be sorted based on a variety of criteria
- Select records to be reported based on a variety of criteria
- Customize report formats

The creating of a Report Writer report is a 2-step process:

- Records are selected for sorting and sorted
- Sorted Records are reported

By default, the Report Writer performs both of these steps for you automatically. However, in situations where you have defined multiple reports to run using the same sorted data, you can perform each of these steps separately. In this way, the sorting of data need only occur once (step #1 above). This sorted data can then be input to multiple reports (step #2 above, repeated as many times as there are reports).

## The Report Definition

A Report Definition is an ASCII (text) file that is created and maintained outside of BOOMS. A text editor (not included with BOOMS) is required to create and maintain a Report Definition. Each Report Definition consists of the following sections:

- **HELP Section** - contains descriptive information that the user can display (Shift-F1) to obtain additional help.
- **VARIABLENAMES Section** defines a report title and the descriptive name for screen variables.
- **VARIABLEDEFINITION Section** defines any variables (e.g. numeric totals that are used in the Report Definition).
- **HEADER Section** defines the format of the report headers
- **DETAILLINE Section**  
**DETAILLINE2 Section**  
**SUBTOTALSLINE Section**  
**GRANDTOTALSLINE Section** these sections define, respectively, the format of a detail print line, the format of an alternate detail print line, a subtotals print line, and the grandtotals print line.
- **SETUP Section** contains optional LOGIC used to initialize the starting search "key" for the PRIMARY FILE. Other variables that require one-time initialization (e.g. printer characters-per-inch and lines-per-inch) can also be set here instead of in the "PRINT SECTION" (see below).
- **RECORDSELECTION Section** contains the LOGIC that is used to select records for sorting.
- **SORTDEFINITION Section** defines the elements that are to be used to sort records.
- **PRINT Section** contains the LOGIC that is used to select records for printing (the print section only reads

SORTED records (i.e. only those records previously selected and sorted).

- **PRINTEOF Section** contains the LOGIC to be applied after the last record is selected for printing (e.g. print final SUBTOTALS and GRANDTOTALS).

Figure 27.1 shows a report definition to print canceled Orders. It is not intended to be run as is (although it could be), but as an example of how the various sections are combined to form a report definition.

```

////////////////////
HELP SECTION:
////////////////////
This report lists CANCELED ORDERS.
Vars:  START/END Order Numbers (optional)
////////////////////
VARIABLENAMES SECTION:
////////////////////
SET &RPTTITLE = 'ORDERS FILE (CANCELED ORDERS)
SET &NWNUMBR2 = 'START ORDER NUMBER (OPT) '
SET &NWNUMBR3 = 'END ORDER NUMBER (OPT) '
////////////////////
VARIABLESDEFINITION SECTION:
////////////////////
&TOTALS(C,7)  VALUE(TOTALS:)
&OQTY(W,4)
&TQTY(W,4)
&ERRMSG1(C,38)  VALUE(START/END ORDER NUMBERS ARE INVALID ..)
////////////////////
HEADER SECTION:
////////////////////
&CDATE                      Broker/Owner Orders Management System          PAGE &PN
&CTIME                      --- ORDERS FILE (CANCELLED ORDERS) ---
////////////////////
ORDER#      BCODE  BROKER NAME                      MCODE  LIST#  QUANTITY
////////////////////
DETAILLINE SECTION:
////////////////////
FILEID(OR)
ORDERNUMBER,4
BROKERCODE,2
BROKERCODE(28),2 EXPAND
MAILERCODE,3
LISTNUMBER,2
&OQTY                      MASK(9999,99X)
CRLF,1
////////////////////
GRANDTOTALSLINE SECTION:
////////////////////
&TOTALS,55
&TQTY,1                      MASK(9999,99X)
CRLF,1
////////////////////
SETUP SECTION:
////////////////////
SET &KEYNUM = 0
SET &KEYSEGO = &NWNUMBR2
IF &NWNUMBR3 = 0
  SET &NWNUMBR3 = 999999
ENDIF
IF &NWNUMBR2 NE 0 AND &NWNUMBR2 GT &NWNUMBR3
  SET &STOPMSG = &ERRMSG2
  STOP
ENDIF
////////////////////
RECORDSELECTION SECTION:
////////////////////
IF ORDERNUMBER GT &NWNUMBR3
  STOP
ENDIF
IF NOT ORDCANCEL
  REJECT
ENDIF
////////////////////
SORTDEFINITION SECTION:
////////////////////
ORDERNUMBER,4,A
////////////////////
PRINT SECTION:
////////////////////
IF &DTLCOUNT > 56
  EJECT
ENDIF
SET &OQTY = RENTQTY-SHIP + EXCHQTY-SHIP
SET &TQTY = &TQTY + &OQTY
////////////////////
PRINTEOF SECTION:
////////////////////
GRANDTOTALS

```

Figure 27-1 Sample Report Definition

## HELP Section

This section consists of up to 21 lines of text that can be customized to provide report-specific information. It is displayed when you press the "RptHelp" Key (Shift-F1) when in the "Variables" portion of the screen. If no HELP section is defined and you

press "Shift-F1", the message "NO HELP AVAILABLE" is displayed. To return to the main screen, press "F9" (Return).

## VARIABLENAMES Section

By its nature, the Report Writer is "generalized". By default, the "field names" applied to screen variables are "generic". That is, without any particular knowledge of the actual report being run, the Report Writer assigns names to screen variables that match the names of the actual variables. Thus, for example, the screen variable that applies to "&WDATE1" is called "&WDATE1". The VARIABLENAMES Section allows you to customize the screen so that the "field names" are more meaningful to the report being run. For example, if screen variable "&WDATE1" is actually a "Mail Date", then you can show this on screen by including the following statement in this section:

```
SET &NDATE1 = ' MAIL DATE '
```

Any of the screen variables can be customized in this way. The Report Writer contains a built-in "field name" variable for each built-in "screen variable". The names are the same, except that the "field name" variables are prefixed with an "N". For example, the "field name" variable for "&WNUMBR1" is "&NNUMBR1" and the "field name" variable for "&WCODE1" is "&NCODE1".

In addition, a Report Title can be set using the built-in variable "&RPTTITLE". This title will appear on screen after a report is selected. Following is an example of setting a Report Title:

```
SET &RPTTITLE = 'LIST USAGE REPORT [IN MAILER SEQUENCE]'
```

 **Only "SET" statements are meaningful in the VARIABLENAMES Section. If the section exists, the screen will be customized to include only those "field names" indicated. You will not be allowed to enter data into the other fields.**

## VARIABLESDEFINITION Section

A "variable" is a name that represents a value that can change from one report to the next. For example, "the total number of Orders" between two dates is a "variable" because this value will change depending upon the 2 dates selected (In the same manner, the two dates that make up the "date range" could also be considered "variables" as they would also change from one report to the next). Before a "variable" can be used in a report (in any of the other sections), it must be defined. Some variables are predefined by the Report Writer (see Predefined Variables). Any other variables that are required for a particular report, must be defined in the VARIABLESDEFINITION SECTION.

Each entry in this section is entered in the following format:

```
&vname(t,l) VALUE(initial-value)
```

where:

**&** - is entered as shown and identifies the name as a VARIABLE.

**vname** - is the name of the "variable". This name cannot exceed eight (8) characters (e.g. TOTORDS or BEGDATE)

**t** - is the one-character variable's "TYPE" as follows:

<b>C</b>	<b>a character type</b>
<b>A</b>	<b>an alpha type (letters are converted to uppercase; otherwise the same as character)</b>
<b>W</b>	<b>a word type (numeric value used for computation: e.g. a Quantity)</b>
<b>I</b>	<b>an integer type (numeric value not used in computation: e.g. an Order Number)</b>
<b>D</b>	<b>is a date type</b>

**l** - is the one to three digit variable's "LENGTH". (The length for a "W" or "I" type can be 2, 4 or 6; length for "D" type must be 2)

**VALUE(initial-value)** - is used to initialize the variable to a particular value.

The following three examples define and initialize a "numeric" variable, a "character" variable and a "date" variable respectively:

```
&TOTORDS(W,2) VALUE(100)
&TOTC(C,6) VALUE(TOTALS)
&COMPDATE(D,2) VALUE(12/31/87)
```

## HEADER Section

The HEADER SECTION defines the format of the “headers” that are to appear at the top of each page (the 1st page and whenever an “EJECT” or “NEWREPORT” action is encountered; see PRINT SECTION). A Report header consists of text and variables (type “C” or “A” only) that can be freely intermixed. Typically, the header defines both a “general” header and “column” headers for detail information to appear in the report. Each header line is printed as it appears in this section (including “blank” lines). That is, the way you see the text on a screen (or a listing) is the way it will print on the report. Each header line has a maximum length of 255 characters.

## DETAILLINE Section & DETAILLINE2 Section

The DETAILLINE SECTION defines the primary BOOMS file being used for the report. The following statement **MUST** be the first statement in this section (See Appendix F for the valid “fids”):

### **FILEID(fid)**


The main function of the DETAILLINE SECTION is to define the format of each detail line that is to appear on the report. A detail line is printed every time a record is selected for printing. (Elements in the “DETAILLINE2 SECTION” usually represent an alternate print format and are only printed when requested by a “PRINTDETAIL2” action). The definition consists of a series of statements that define the elements to be printed. Each statement is in one of the following format (Brackets “[..]” indicate option parameters):

```
[ fid.] element(l,o),s [ MASK(maskimage) ] [ EXPAND ]
SPACE,n
CRLF,n
zCRLF,n
EJECT
```

where:

**fid** - is the FILE-ID, if it is other than the Primary File. The Report Writer can access other (secondary) BOOMS files once a connection has been made to them. This connection is made by using the EXPAND keyword for an element on the primary BOOMS file.

**element** - is the “item” to be printed. An element is either a DICTIONARY element (See appendix F) or a VARIABLE.

 **If the first character of a print element is a “0FFh” (hexadecimal highvalue or -1), the printing of the element will be suppressed.**

**l** - is the length of the element. It defaults to the defined length for the DICTIONARY ELEMENT or VARIABLE

 **If the length of a print element is zero (0), the element will not be printed. This is useful if you are just defining an element so that it can be EXPANDED to connect to a related file.**

**The maximum length of an element is 255 characters. If the element exceeds this length (e.g. when printing “raw” records) it must be broken down into multiple records.**

**o** - is the offset, within element, where printing is to begin. It defaults to zero (printing begins with the 1st position)

**s** - is the number of trailing blanks to be printed following this element. The default is NO SPACES.

**MASK(maskimage)** - is the definition of the way that a numeric element should be printed. Each DICTIONARY element has a default “maskimage” (which can be overridden here). A “maskimage” **MUST** be specified for a numeric VARIABLE. The “maskimage” consists of numeric place holders (the number “9”), a significance start indicator (the letter “X”) and editing characters (commas, decimal place, dollar sign, etc). For example, an “order number” could be indicated with the following “maskimage” (because there is no significance start indicator, all 6 digits will print, including leading zeroes):

**MASK(999999)**

A dollar amount could be indicated with one of the following “maskimages”. The first would print “blanks” if the amount were 0, the second would print “.00” if the amount were 0 (leading zeros and editing characters are suppressed in both cases):

**MASK(99,999.9X)**

**MASK(99,99X.99)**

If the first position of the “maskimage” is an “editing character”, it is treated as follows:

A **dollar sign [ \$ ]** will float to the position preceding the 1st significant character.

A **Left Parenthesis [ ( ]** will cause unconditional parentheses to be placed around a number (if the number is not to be printed as a "blank").

**Any other "editing character"** will fill the field up to the first significant character.

**EXPAND** - is a special KEYWORD that causes the printing of a related DICTIONARY element instead of the indicated element. This is most often used to print "NAME" information instead of "CODED NAME" information (e.g. print a Mailer's Name instead of a Mailer's Code. See Appendix F for the fields that can be EXPANDED). Once an element has been EXPANDED, the entire secondary BOOMS file record is available for printing (using its "fid" or file id).

**SPACE,n** - is used to force spaces between elements ("n" is the number of Spaces). Typically this would be used to put spaces at the beginning of a detail line.

**CRLF,n** - is used to force spaces between lines ("n" is the number of Carriage Return, Linefeeds).

**zCRLF,n** - is used to do a "conditional Carriage Return, Line Feed" when NOT forcing "TABS/SPOOL" ("n" is the number of Carriage Return, Linefeeds). Many reports were not designed with the idea that they would be converted to a tab-delimited format. For example, reports such as ORDRPRTE (Mailer Usage) contain dissimilar multi-line printouts that is not easily put into a tabbed format. To accommodate such reports, the multiple lines can be combined by making use of a "conditional line feed" (i.e. instead of a line feed, a tab is inserted). **PLEASE NOTE:** Lissan Computing Co. will not be modifying distributed reports to make them 100% compatible with this feature. However, if you desire report customization, this service can be provide for an additional fee.

**EJECT** - is used to force a new page (Formfeed). Also, before the next detail line is printed, HEADERS will be reprinted.

## SUBTOTALSLINE, SUBTOTALSLINE2 & GRANDTOTALSLINE Sections

The SUBTOTALSLINE, SUBTOTALSLINE2 and GRANDTOTALSLINE Sections define lines that are to be printed when a SUBTOTALS, SUBTOTALS2 or GRANDTOTALS keyword is encountered in the PRINT SECTION or PRINTEOF SECTION. Statements in these sections are in the same format as the DETAILLINE SECTION. Typically, only summary information is formatted for print in these sections.

## SETUP Section

The SETUP SECTION provides optional LOGIC for PRIMARY FILE starting "key" selection and one-time initialization of variables (e.g. printer LPI and CPI variables). By default, the Report Writer begins the record selection process with the first record on the PRIMARY File. However, this may not be appropriate in all situations. For example, a report that is setup to summarize Accounts Receivable information has no need to look at Accounts Payable records. If no "key" setup is provided, you could include logic in the RECORDSELECTION SECTION to REJECT Accounts Payable records as they are read from the Accounting File. This can be time consuming. An alternative is to tell the report writer to automatically skip directly to the first Accounts Receivable record.

The information needed by the Report Writer to accomplish initial "key" (index) positioning is obtained from the following "Predefined" Variables (See "&KEYNUM/&KEYSEGN Combinations by FILE ID"):

- **&KEYNUM**
- **&KEYSEG0**
- **&KEYSEG1**
- **&KEYSEG2**
- **&KEYSEG3**

The general formats of statements in the SETUP SECTION are:

SET variable = element

IF condition  
action  
ELSE  
alternate action  
ENDIF

In the first format, "variable" is a name defined in the VARIABLESDEFINITION SECTION (or a "Predefined Variable"). "element" can be another variable (or part of one), a "constant", or (in the case of a "numeric variable"), an arithmetic expression (e.g. &ABC + &XYZ). A "constant" can be a "character constant" (enclosed in quotes) or a "numeric constant" (not enclosed in quotes). DICTIONARY elements are not allowed in the SETUP SECTION. Following are examples of valid SET statement:



```
SET &KEYNUM = 0
SET &KEYSEG0 = 10000
SET &CPI = 16
SET &LPI = 8
```

In the second format, "IF...ELSE...ENDIF" logic is used to conditionally SET variables. "condition" is the LOGICAL criteria to be tested. All of the following are examples of valid conditional structures (See "Valid Arithmetic and Logical Connectors"):

```
&A = &AB
&A NE &B
&KEYSEG0 = &WNUMBR1
&A = &B + &C
&WDATE1 = 12/31/87
```

"Action" is:

**SET** - valid SET statement as described above

**STOP** - indicates that the report should be terminated (presumably because required screen variables have not been entered properly). If this occurs, you will receive error message "FORCED STOP DURING REPORT SETUP". (See built-in variable "&STOPMSG").

The "ELSE...alternate action" is optional. If not specified, no alternate action is taken by the Report Writer. Valid alternate actions are the same as "action" above. Following are examples of valid IF...ELSE...ENDIF statements:

```
IF &WCODE1 = ' '
  SET &WCODE1 = 'R'
ENDIF

IF &WDATE1 = &NULDATE
  SET &WDATE1 = '01/01/88'
ENDIF
```

Following are 2 examples of using the "&KEY" variables. In example (1), Accounts Receivable or Accounts Payable records are to be selected from the Accounts file (as indicated by an "R" or "P" in screen variable &WCODE1). Variable "&HEX000" is predefined as a binary zero. In example (2), Orders for a particular mailer for a particular mail date are desired (as indicated by screen variables &WCODE1 and &WDATE1). Note that in example (2) because "&KEYSEG1" is being set to a "date variable", it must be initialized with a "date template".

```
SET & KEYNUM = 0
SET &KEYSEG0[1] = &WCODE1
SET &KEYSEG0[1,1] = &HEX000

SET & KEYNUM = 3
SET &KEYSEG0 = &WCODE1
SET &KEYSEG1 = ' - - '
SET &KEYSEG1 = &WDATE1
```

## RECORDSELECTION Section

The RECORDSELECTION SECTION provides the LOGIC for initial record selection (i.e. record to be input to the SORT). Each record read from the primary file (as indicated by the FILEID keyword in the DETAILLINE SECTION) is passed through the RECORDSELECTION logic. Records can be selected based on a variety of criteria, including:

- records that contain a certain code
- records that contain a date in a specific range
- records that contain a numeric field (e.g. quantity) that is greater than a certain amount
- stop selecting records when a certain condition is met

The general formats of statements in the RECORDSELECTION SECTION are:

```
SET variable = element
IF condition
```

```

    action
  ELSE
  alternate action
ENDIF

```

In the first format above, "variable" is a name defined in the VARIABLESDEFINITION SECTION. "element" can be another variable (or part of one), a DICTIONARY element (or part of one), an "expanded" dictionary element, a "constant", or (in the case of a "numeric variable"), an arithmetic expression (e.g. &ABC + &XYZ or AMOUNTDUE - AMTPAID). A "constant" can be a "character constant" (enclosed in quotes) or a "numeric constant (not enclosed in quotes). In the RECORDSELECTION SECTION, "SET variable" can be used to obtain a value for conditional testing or to be used in the SORTDEFINITION SECTION (or even the PRINT SECTION or DETAILLINE SECTION). Following are examples of valid SET statements:

```

SET &CURRAMT = AMOUNTDUE - AMTPAID + 1000
SET &TOTC = 'TOTALS:'
SET &TYPE = 'A/R'
SET &PREVCODE = BROKERCODE
SET &NUMBR = TRANSDESCRIP[6,1]
SET &ACCTCODE = TRANSACCOUNT[E]
SET &HICODE = MAILERCODE[3]

```

 **A SET can be used to convert a number (e.g. ORDERNUMBER) to a character-based number (i.e. Convert ASCII to Number and visa versa). You may find this useful if you are trying to print a numeric variable in a header.**

**If you set an "Alpha" field type from a "character" field type, the field is converted to UPPERCASE.**

**If the right side of a SET statement indicates an element of length "255", it will be translated as a length of "512". For example, in the statement "IF &WCODE2 @NOTIN RAWREC[255]", the length of RAWREC will actually be "512". This is necessary since the maximum number that can be specified is "255".**

In the second format above, "IF...ELSE...ENDIF" logic is used to conditionally SELECT records, SET variables, REJECT records, or STOP the record selection process. "condition" is the LOGICAL criteria to be tested. All of the following are examples of valid conditional structures (See "Valid Arithmetic and Logical Connectors" for more information):

```

&A = &AB
&A NE &B
ORDCANCEL
ACCTVOID AND ACCTPIF
&A LT &B
&A GT &B
&A = &B + &C
ORDERDATE = 10/10/87
IF &ZIP @IN SELECTION1
IF &CODEA1 @INF CATEGORYCODS

```

"Action" is:

**SELECT** to select a record without any further testing.

**REJECT** to reject a record without any further testing.

**STOP** to stop the record selection process and proceed immediately to the SORT

**SET** is a valid SET statement as described above

**REPOSITION** to reposition the primary input file. This action would normally occur after resetting one of the primary file's "&KEYSEG's". That is, repositioning is accomplished in the same manner as initial file positioning in the SETUP SECTION. RECORDSELECTION logic is repeated for the newly positioned file, so any statements after REPOSITION are ignored for the current record (unless the variable, &BREPOS, is set to "YES" which will cause the repositioning to occur after all logic for the current record is complete).

**GETEXTERNAL** - reads a record (&EXTREC) from the External File (&EXTFILE). The read can be checked for successful completion by checking variable &EXTERR as follows

"blank" (IF &EXTERR = ' '), the read was successful.

"E" (IF &EXTERR = 'E'), then the file has been completely processed (end-of-file).

"F" (IF &EXTERR = 'F'), then the "External File" is not specified. Any other value indicates an error.



**PAUSE** causes the Report Writer to “pause” its operation. When a pause action is encountered, the screen is redisplayed (all variables are redisplayed) and the cursor is positioned in the “Variables” portion of the screen. Although “PAUSE” is intended to be used as a development tool, it is possible to design a report that allows a certain amount of interaction with the person who is running the report. When a “PAUSE” occurs, you will receive status message:

**PAUSED...[Enter Values in Window] .....**

You can continue with the report by pressing “Esc” (or by filling in the variables as desired).

 **The “F4” key can cause a “PAUSE” on demand.**

The “ELSE...alternate action” is optional. If not specified, no alternate action is taken by the Report Writer. Valid alternate actions are the same as “action” above. Following are examples of valid IF...ELSE...ENDIF statements:

```
IF &SRCOUNT = 100
  STOP
ENDIF

IF ACCTPIF or
  ACCTVOID or
  ACCTFREE or
  AMOUNTPAID = AMOUNTDUE
  REJECT
ENDIF

IF &A = &AB
  SET &A = &A + &B
ELSE
  SET &A = &A - &B
ENDIF

IF ORDERNUMBER = 3522
  SET &KEYSEG0 = 7400
  REPOSITION
ENDIF
```

 **If no “action” is indicated after all logic has been done, the record will be SELECTED. If “Esc” is pressed during the record selection process, end-of-file (EOF) is signaled and the SORT is immediately executed.**

## SORTDEFINITION Section

The SORTDEFINITION SECTION is used to define how records selected in the RECORDSELECTION SECTION are to be sorted. Following are valid fields for sorting:

- **DICTIONARY** elements from the primary file
- **EXPANDED** fields
- **Variables**

Valid names (DICTIONARY Elements) for fields in the primary file (as indicated by the FILEID keyword in the DETAILLINE SECTION) are defined in Appendix F. EXPANDED fields are typically the name associated with a Code or Number (e.g. the Broker Name or the List Name). Variables have their values “SET” in the SETUP SECTION or RECORDSELECTION Section. Following is the format of each statement in the SORTDEFINITION Section:

**element[E],I,o**

where:

**element** - is a DICTIONARY ELEMENT or a VARIABLE

**[E]** - is optional and indicates that a DICTIONARY ELEMENT is to be EXPANDED before Sorting.

**I** - is the optional length of the field. It defaults to the defined length (from the DICTIONARY or VARIABLE definition). If this is an EXPANDED element, the length refers to the length of the EXPANDED element.

**o** - is the sort order: either “A” for ascending or “D” for descending. The default is ascending.

Following are valid sort statements:

```
ACCTTYPE,,D
ACCTCODE[E],,A
```

In the above example, Accounting records are being sorted. ACCTTYPE is a one character DICTIONARY ELEMENT ("R" = Receivables, "P" = Payables). ACCTCODE is a DICTIONARY ELEMENT indicating the code for an account. The EXPANDED value is the 32-character name of the Account. The two statements cause the Accounting File to be sorted with Receivables (R) before Payables (P) and in alphabetical sequence by Account Name (not code).

```
ORDERNUMBER,,A
```

```
TRANSDATE,,A
```

```
TRANSNUMBER,,A
```

In the above example, records on the Current period Journal are being sorted in ascending sequence by Order Number, Transaction Date, and Transaction Number.

## PRINT Section

The PRINT SECTION provides the LOGIC for record printing that are output from the SORT. For example, records can be printed that contain a certain code, contain a date in a specific range, or contain a numeric field (e.g. quantity) that is greater than a certain amount.

The general formats of commands in the PRINT SECTION are:

```
variable = element
```

```
IF condition
```

```
  action
```

```
ELSE
```

```
  alternate action
```

```
ENDIF
```

In the first format above, "variable" is a name defined in the VARIABLESDEFINITION SECTION. "element" can be another variable, a DICTIONARY element, a "constant", or, in the case of a "numeric variable", an arithmetic expression (e.g. &ABC + &XYZ or AMOUNTDUE - AMTPAID). A SET variable can be used to obtain a value for conditional testing or for printing.

Following are examples of valid SET statements:

```
SET &CURRAMT = AMOUNTDUE - AMTPAID
```

```
SET &TOTC = 'TOTALS:'
```

```
SET &TYPE = 'A/R'
```

```
SET &PREVCODE = BROKERCODE
```

```
SET &NUMBR = TRANSDESCRIP[6,1]
```

```
SET &ACCTCODE = TRANSACCOUNT[E]
```

```
SET &HICODE = MAILERCODE[3]
```

```
SET &NEWDATE = &OLDDATE + 30
```

```
SET &NEWDATE = &OLDDATE - 30
```

 **A SET can be used to convert a number (e.g. ORDERNUMBER) to a character-based number (ASCII to Number and visa versa). You may find this useful if you are trying to print a numeric variable in a header.**

**If you set an "Alpha" field type from a "character" field type, the field is converted to UPPERCASE.**

In the second format above, "IF....ELSE....ENDIF" logic is used to conditionally SELECT records, SET variables, REJECT records, print SUBTOTALS, print GRANDTOTALS, or STOP the record printing process. "condition" is the LOGICAL criteria to be tested. All of the following are examples of valid conditional structures (See Valid Arithmetic & Logical Connectors for more information) :

```
&A = AB
```

```
&A NE &B
```

```
ORDCANCEL
```

```
ACCTVOID AND ACCTPIF
```

```
&A GT &B
```

```
&A = &B + &C
```

```
ORDERDATE = '10/10/87'
```

```
IF &ZIP @IN SELECTION2
```

IF &CODEXX @INF CATEGORYCODS

“Action” is:

**SELECT** to select a record without any further testing

**REJECT** to reject a record without any further testing

**STOP** to stop the record printing process

**SET** a valid SET statement as described above

**PRINTDETAIL** to print a detail line in the middle of PRINT logic. Additional logic can still be applied to the primary record after it has been printed.

**PRINTDETAIL2** to print an alternate detail line format (as defined in the DETAILLINE2 SECTION) in the middle of print logic. Such an action could be used, for example, to print a variable number of selections for an Order.

**SUBTOTALS** - causes the printing of elements as defined in the SUBTOTALSLINE SECTION.

**GRANDTOTALS** causes the printing of elements as defined in the GRANDTOTALSLINE SECTION.

**NEWREPORT** causes the page number variable (&PN) to be reset to 1 and the linecount variables (&LCOUNT and &DTLCOUNT) to be reset to 0. Also, HEADERS will be reprinted before the printing of the next detail line. An EJECT, however, IS NOT performed.

**EJECT** causes a Form Feed on the printer and the linecount variables (&LCOUNT and &DTLCOUNT) to be reset to 0. Also, HEADERS will be reprinted before the printing of the next detail line.

**PAUSE** causes the Report Writer to “pause” its operation. When a pause action is encountered, the screen is redisplayed (all variables are redisplayed) and the cursor is positioned in the “Variables” portion of the screen. Although “PAUSE” is intended to be used as a development tool, it is possible to design a report that allows a certain amount of interaction with the person who is running the report. When a “PAUSE” occurs, you will receive status message:

**PAUSED...[Enter Values in Window] .....**

You can continue with the report by pressing “Esc” (or by filling in the variables as desired).

 **The “F4” key can cause a “PAUSE” on demand.**

The “ELSE...alternate action” is optional. If not specified, no alternate action is taken by the Report Writer. Valid alternate actions are the same as “action” above. Following are examples of valid IF...ELSE...ENDIF statements:

```
IF &PREVCODE NE ACCTCODE
  SUBTOTALS
  SET &PREVCODE = ACCTCODE
ENDIF
```

```
IF ACCTBROKER
  SET &ACCTTYPE = 'B'
ELSEIF ACCTOWNER
  SET &ACCTTYPE = 'O'
ELSEIF ACCTSERVICEB
  SET &ACCTTYPE = 'S'
ENDIF
ENDIF
ENDIF
```

 **If no “action” is indicated after all logic has been done, the record will be PRINTED.**

## PRINTEOF Section

The PRINTEOF SECTION contains print logic to be done when end-of-input is detected on the Sorted file (the last record has been processed). The format of statements in this section is the same as those in the PRINT SECTION. Typically, this section is used to perform final “SUBTOTALS” and “GRANDTOTALS” processing.

## Report Writer Selection

The Report Selection screen is selected by “Function=R” from the signon screen or by “Function=R2” from the Main Selection Menu. This screen is divided into 2 sections (see Figure 27.2 and Figure 27.3):

- The Top section which contains the “Report Description File, the “File Override” and the “Report Title”
- The Window section which contains the “screen VARIABLES”. If the Report Description File does not contain

a VARIABLENAMES SECTION, you will see “raw” variable names (Figure 27.2). Otherwise, you will see only those variable names defined for the report (Figure 27.3).

Screen variables give you the ability to dynamically modify the logic of the report without having to change your Report Definition (they can be viewed as being similar to the ADDITIONAL QUALIFICATION for “Orders” reporting; See Chapter 4). These “variables” are provided for you convenience. They are not required for the Report Writer to function properly. In addition, the “field names” shown on screen can be modified (see the VARIABLENAMES Section).

A second screen (Figure 27.4), accessed by pressing F7 (DispFil) is used to display (or Update) “Files Information”, which is used to tell the Report Writer (to return to the first screen, press “F7” (DispVar) again):

- The name and location of the file used to contain records selected for sorting (file created by the record selection process).

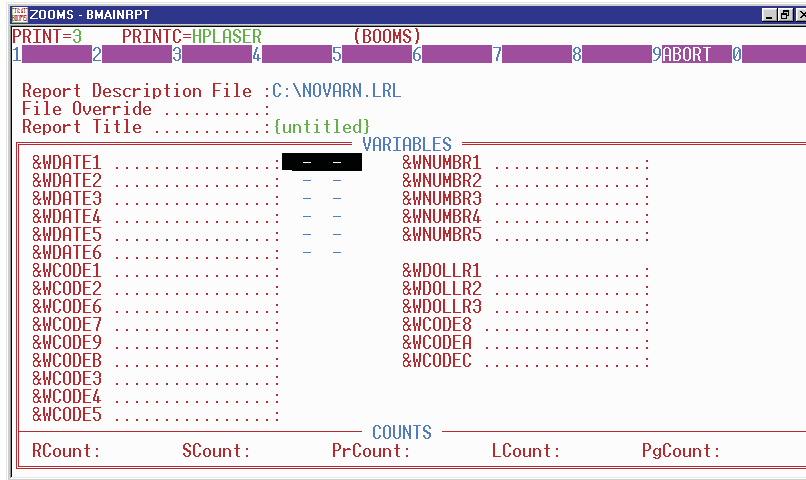


Figure 27.2. Report Writer Screen (showing “raw” Variable Names)

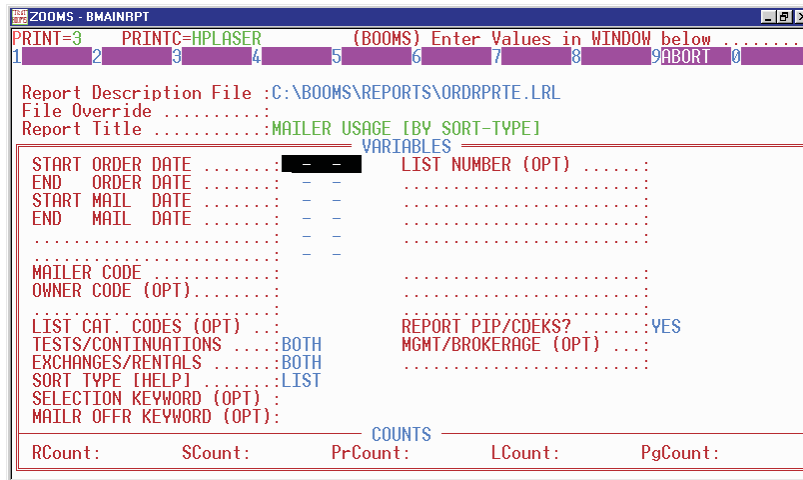


Figure 27.3. Report Writer Screen (showing “custom” Variable Names)

- The name and location of the file used to contain sorted record pointers (file input to the print process).
- The drive used to contain temporary sort files (Sortwork drive)
- The SortSummary Table size (Used for SORTSUMMARY “Rankings”)
- The “External File” location and name (typically used for 2-step reporting)
- The “Force TABS/SPOOL” setting (Y/N) for automated report conversion to a “tab-delimited file. Such a file **MAY** be appropriate for import into programs such as Excel (**Note:** the variable **Force** “TABS/SPOOL” can **NOT** be set while the cursor is in the VARIABLES Section; also, it resets itself to “N” for each new report). The

best way to insure that output from the Report Writer is appropriate for import to any external program would be to design it for that purpose. However, you may find the “Force TABS/SPOOL” technique useful for a number of the distributed reports. If you have a particular report that you are interested in creating, you should also create a TAB-delimited column-header file that is appropriate for your needs (same name as report file , but with an ex-

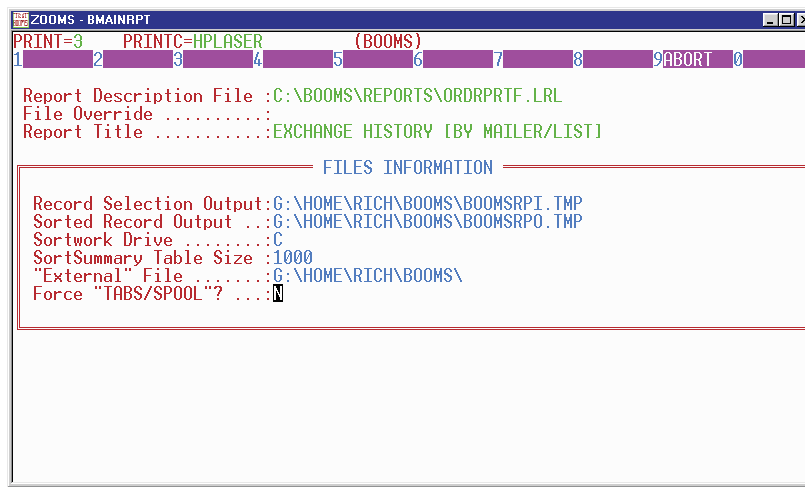


Figure 27.4. Report Writer “Files Information” Screen

tension of “TXT”) or design a column header template (e.g in Excel). You may need to experiment to determine the proper placement of the TABS. Setting this value to “Y” causes a report to assume the following characteristics (see Predefined Variables for more information):

**&PCONTROL = N**  
**&DLIMITER = T** and  
**&PRINT = S**

In addition to fill-in fields, the Report Selection screen also displays the following 5 counters. These values are displayed at the end of record selection, the end of printing and whenever you “pause” a report:

- **RCount:** a count of the number of records read during the RECORD SELECTION process.
- **SCount:** a count of the number of records selected during the RECORD SELECTION process.
- **PrCount:** a count of the number of DETAIL records printed during the PRINT process. This is the same number that appears in the status message when the report is finished
- **LCount:** a count of the total number of lines printed during the PRINT process.
- **PCount:** a count of the total number of pages printed during the PRINT process.

## Report Description File

The Report Description file is an ASCII (text) file that is created and maintained using a “text editor” outside of BOOMS. It consists of various sections that define the format and contents of a particular report (See Figure 27.1). When filling in this field, specify the full “drive” and “path name” for the location of the file. For example, if the Report Description file is named “ACCTPRT1.LRL” and is located in subdirectory “\BOOMS\REPORTS” on drive “D”, then enter the following for “Report Description”:

**D:\BOOMS\REPORTS\ACCTPRT1.LRL**

 The default “path” for the description file can be specified through a global parameter (See Chapter 24).

The description file can be entered without the “.LRL” extension. The Report Writer will automatically append “.LRL” if no file extension is given.

After you have filled in the “Report Description File” field (and optionally the “File Override”), the cursor will move to the Window section of the screen and you will receive status message:

Enter Values in Window below .....

## File Override

The "File Override" allows you to select the primary FILE-ID (FID) when multiple files are mapped to the same Dictionary. At this time, this only applies to the Current Period Journal (FID=JO), the Period-End Journal (FID=JP) or any of the Journal History files. By using this field, a single report could be developed to access any of these files. For example, if you normally run your "Journal" reports at month-end (last day of the month), any reports to access the Journal would normally refer to FID=JO. If on occasion you must run the reports at the beginning of the following month (after you have begun to close out the books for the prior month and thus created a new Journal), you can use a "File Override" of "JP" to access the Period-End Journal (the Journal from the prior month).

You can also enter an actual file name for a "File Override". For example, if the Journal history files are in directory "\BOOMS\DATABASE" on drive F, you could enter "F:\BOOMS\DATABASE\BJ010731.DBF" to access the July 2001 Journal or F:\BOOMS\DATABASE\BJ2001.DBF to access a combined journal for 2001 (assuming you've created a combined journal for 2001).

## Report Title

If the report that you are running has a title, it will appear in this field after you have filled in the "Report Description File" and, optionally, the "File Override". Otherwise, this field will show "{untitled}".

## Record Selection Output File

The Record Selection Output File is used to contain records selected for sorting. Each time a record is selected, the Report Writer builds a record (to be input to the SORT). This record consists of:

- The address of the record on the HardDisk (4 characters)
- Each field that is selected for sorting.

For example, if you are sorting the "Orders file" based on "Order Date" (2 characters) and "Order Number" (4 characters) sequence, the Report Writer builds selected records with the following format:

Location of the full Order record - 4 characters

Order Date - 2 characters

Order Number - 4 characters

Thus, if all records from a 20,000-order "Orders file" are selected, the size of the Record selection file will be:  $20,000 \times 10 = 200,000$  characters.

When filling in this field, specify the full "drive" and "path name" for the location of the file. In general, because this is an intermediate file (it is used internally by BOOMS), you should take the default name for the file (d:\BOOMS\RPI.TMP, where "d" is the drive letter specified for DOS FILES when BOOMS was started). This field should be left "blank" to skip the record selection/sorting process. This can be done only if a "Sorted Record Output File" (see below) has been previously created. This technique is useful if you have multiple reports that run off the same sorted data.

## Sorted Record Output File

The Sorted Record Output File contains the records that are output from the Sort (and input to the record printing process). By default, each sorted record contains only the address portion of the records selected for sorting (4 characters). The Report Writer will use this address to locate the full record on the HardDisk. In order to estimate the size of this file, multiply the number of records selected by 4. Thus, if all records from a 20,000-order "Orders file" are selected, the size of the Sorted Record Output file will be:  $20,000 \times 4 = 80,000$  characters.



**BOOMS will retain the entire sorted record if any of the variables "&SORTFLD1 through &SORTFLD9" are used in the PRINT SECTION. This allows access to the actual sort fields during the printing process.**

When filling in this field, specify the full "drive" and "path name" for the location of the file. In general, specify a different name if you expect to reuse this file for additional reports that can be run using the same sorted data. Otherwise, you should



take the default name for the file (d:\BOOMS\SRPO.TMP, where 'd' is the drive letter specified for DOS FILES when BOOMS was started).

For example, if you want the Sorted Record Output File to be named "ACCTDATA" and located in subdirectory "\BOOMS\SORTDATA" on drive "D", then enter the following for "Sorted Record Output":

**D:\BOOMS\SORTDATA\ACCTDATA**

 **The Report Writer will not do any cleanup of the files that you specify. It will however "reuse" files. So if you specify the same name as an existing file, the old file will be overlaid with the new.**

## Screen Variables

The screen variables are:

- **&Sortwork Drive** - by default, the Disk Drive that BOOMS uses as a "Sort Workarea" is the one that is specified in the startup parameter "/W:". This drive, which is usually a local hard disk, is used by "OPT-TECH SORT" during the sorting process. The default drive is usually more than adequate for workarea purposes. However, it is possible to improve sorting performance slightly by using a Ram Disk as a sortwork drive.
- **&SortSummary Table Size** - represents the size of the area used for performing Summary Rankings (the default is 1000 units or 4K)
- **&WDATE1, &WDATE2, &WDATE3 and &WDATE4** are date variables
- **&WNUMBR1, &WNUMBR2, &WNUMBR3 and &WNUMBR4** are numeric variables (whole numbers)
- **&WDOLLR1, &WDOLLR2 and &WDOLLR3** are numeric dollar amount variables
- **&WCODE1, &WCODE2, &WCODE6, &WCODE7 and &WCODE8** are character variables(8 characters)
- **&WCODE3, &WCODE4 and &WCODE5** are character variables (30 characters)

## Status/Error Lines (If Any)

The Status/Error message area on line 1 will keep you posted on the status of a running report. If any errors occur, you will be informed by a message in this area. Following are the types of status messages you will receive while a report is running ("nnnnn" are running counts of record processed, "dddddd" is a count of detail records):

**Searching in Progress .....**  
**nnnnn Record(s) selected for Sorting ...**  
**nnnnn Record(s) being Sorted .....**  
**nnnnn Record(s) being Merged .....**  
**nnnnn Record(s) Reported (dddddd DTL).**

The bottom line on the screen is used by the Report Writer to display the contents of a statement in the Report Description File that is causing a Report Writer error to occur (the nature of the error is indicated in the STATUS/ERROR MESSAGE AREA). This field is normally only useful during report development. Functional reports should not cause any messages to be displayed here.

If no syntax errors occur during the running of a report, the Report Writer will use this line to indicate how long it took to run the report. The format of the message is:

**RUNTIME=hh:mm:ss**

## Report Pause

While a report is doing Record Selection or Printing, you can manually cause it to "pause" by pressing Function Key 4 (F4). This is equivalent to a "PAUSE" action. When a pause action is encountered, the screen is redisplayed (all variables are redisplayed) and the cursor is positioned in the "Window" of the screen. When a "PAUSE" occurs, you will receive status message:

**PAUSED...[Enter Values in Window] .....**

You can continue with the report by pressing "Esc" (or by filling in the variables as desired).

☞ If you are printing, you may need to “**HOLD DOWN**” the F4" key to cause a “**PAUSE**”.

## Predefined Variables

The following Variables are predefined by the Report Writer and should not be declared in the VARIABLESDEFINITION SECTION.

- **&AUTHSIDS** - is the list of authorized Sales-IDs from the users SigonID record.
- **&BSFNAME** – is a 12-character override for the BOOMS spool file name. Normally, BOOMS generates names of the form “BTEMPnnn.BSF” where “nnn” is a number from 001 to 999. You can use &BSFNAME to set a different name. The following rules apply: (1) the file extension cannot be more than 3 characters (standard DOS); (2) if positions 6-8 are non-numeric (the “nnn” part), BOOMS will NOT attempt to auto-increment the “nnn” to change the file name; (3) If you set a file name and expect BOOMS NOT to change it, do not put all numerics in position 6-8 (e.g if you set a file name of TOYS0901.TXT expecting the name to contain a date of 09/01, BOOMS may change it to TOYS0902.TXT, TOYS0903.TXT, etc.); (4) If you specify a file name that already exists, you will receive error message:

**FILE “ffffffffffffffffffff” Exists (ERROR)**

- **&CURRDATE** - is the current date in BOOMS (internal) format. Used for Date comparisons.
- **&CDATE** - is the current date (18-characters) in “English” format (mmmmmmmm dd, yyyy).
- **&CDAT** - is the current date in “mm-dd-yy” format.
- **&CPI** - is “Characters-per-inch” (Pitch) for the report. The default is 10 CPI.
- **&CRLF** - is a “Carriage Return” and “Line Feed” to be sent to the printer.
- **&CSVAR** can be used to position to a screen variable prior to a pause. For example, **SET &CSVAR =** “WCODE1 “, will position to the &WCODE1 screen variable (Note: &WCODE1 must be an input field).
- **&CTIME** - is the current time in “hh:mm:ss” format
- **&DIFFDAYS** is the difference (in days) between 2 BOOMS dates. The right side of a SET instruction refers to 2 dates when the left side refers to &DIFFDAYS (e.g. SET &DIFFDAYS = &CURRDATE - &PAYDUEDT)
- **&DLIMITER** - can be set to “T” (Tab), “Q” (Quotes/Comma) or “C” (Comma) to automatically delimit an output file. This is useful, for example, if you must send a tab-delimited file to a spreadsheet program. If a report is tab-delimited, each “Alpha” or “Character” field will have trailing “blanks” truncated.
- **&DOSCMD** - can be set to a Dos Command (just like DOSCMD(.....) from the Function Line on a menu). This command would typically be used the SETUP SECTION, perhaps to issue a Netware Capture command to change a printer location.
- **&DTLCOUNT** is the current number of “DETAIL” lines printed on the report. This value can be used to determine when a page should be “EJECTED” based on a count of “DETAIL” lines. An “ EJECT” resets this value to 0.
- **&ESFLAG** is the “end-of-file” indicator for Record Selection. By default, when “end-of-file” is reached during Record Selection, the report Writer automatically proceeds to SORT the selected records. If you set &ESFLAG to “Y”, the Report Writer will return control to the beginning of your “Record Selection Section” with &ESFLAG set to “E”. Your logic can then test for “end-of-file” (IF &ESFLAG = “E” ...) and take appropriate action (perhaps REPOSITION).
- **&EXTFILE** is the 50-character name that relates to the screen variable “External File”. It can be set under program control or entered directed (See GETEXTERNAL action)
- **&EXTREC** is the contents of an “External File Record”. It must contain all ASCII (text) fields that can be broken down using sub-string processing (See GETEXTERNAL action)
- **&EXTERR** is an error indicator for access to &EXTFILE. (See GETEXTERNAL)
- **&FILEOVR** is the “File Override” variable from the Report Writer screen. At this time, this variable is only used to override “Journal” input.
- **&FORCTABS** - value of screen variable “Force TABS/SPOOL?” (Y or N)
- **&FORMLEN** is used to set a form length that is supported by BOOMS. Valid values are “1”, “11”, “5.5”, “3.5”, “U1” or “U2”. These form lengths refer to values set in the current printer definition.

- **&HCONTROL** is a value used to determine if headers are to be printed on a report. The default is “Y” (print headers). Any other value will cause headers to be suppressed.
- **&HEX000** is a 1-character binary value from 1 ( &HEX000[1,0] ) to 255 ( &HEX000[1,255] ).
- **&JUSTIFY** is a 1-character variable that can be set to “L” (default), “R” or “C” to Left Justify, Right Justify, or Center a character variable as indicated by the next SET statement.
- **&KEYNUM** is a value used to determine the sequence in which records from the PRIMARY File are read. See below for valid &KEYNUM specifications.
- **&KEYSEG0** is a value used to determine the starting record to be read from the PRIMARY File. See below for valid &KEYSEG0 specifications.
- **&KEYSEG1** is a value used to determine the starting record to be read from the PRIMARY File. See below for valid &KEYSEG1 specifications.
- **&KEYSEG2** is a value used to determine the starting record to be read from the PRIMARY File. See below for valid &KEYSEG2 specifications.
- **&KEYSEG3** is a value used to determine the starting record to be read from the PRIMARY File. See below for valid &KEYSEG3 specifications.
- **&LCOUNT** is the current number of lines printed on the current report page (both “DETAIL” lines and “HEADER/TOTALS” lines). This value can be used to determine when a page should be “EJECTED” based on a count of all lines printed on a page. An “EJECT” resets this value to 0.
- **&LCOUNTT** is the total number of lines printed for the current run of the report writer. This variable is used by BOOMS to display the “LinCount” field on the screen after the report is finished. This variable should not be modified.
- **&LONGDAT** is automatically set whenever a “character-based” date is set from a “BOOMS” (DOS) date. &LONGDAT is a 10-character field in the following format:

x mm/dd/yyyy

where x = day of the week: 0=Sunday ... 6 = Saturday (followed by a single “blank”)  
 mm = month  
 dd = day  
 yyyy = 4-digit year (1980 - 2079)

For example, if &WDATE1 is a BOOMS date of “02/01/97” and &CDATE is an 8-character-based date field (00-00-00), the statement: SET &CDATE = &WDATE1 would result in a &LONGDATE of “6 02/01/1997”

- **&LPI** is the “Lines-per-inch” for the report. The default is 6 LPI.
- **&LUCODE** is the Licensed User code (your company code).
- **&LUCODEP** is the Licensed User Print code (your company print code). Normally, this is the same as &LUCODE.
- **&LUNAME** is the Licensed User name (your company name).
- **&NMASK &NMASK** – is used to set a numeric mask for a following “number-to-character” SET statement. The length of the mask should be equal to the length of the character variable being set or else truncation will occur. For example, if you wanted to place an edited numeric value (perhaps a quantity) in the header of a report, you could do something like:

```
SET &NMASK = '999,99X'
SET &HQTY = &QTY
```

where &HQTY is a 7-character “character variable”  
 where &QTY is a “doubleword numeric variable”

If &QTY contains a value of 50000 then &HQTY would contain “ 50,000”.

- **&NWDAT1 ... &NWDAT6** are the 25-character display “field names” for &WDATE1 ... &WDATE4
- **&NWNUMBR1 ... &NWNUMBR5** are the 25-character display “field names” for &WNUMBR1 ... &WNUMBR5
- **&NWDOLLR1 ... &NWDOLLR3** are the 25-character display “field names” for &WDOLLR1 ... &WDOLLR3

- **&NWCODE1 ... &NWCODE8** are the 25-character display “field names” for **&WCODE1 ... &WCODE8**
- **&PCOUNT** is the total number of pages printed for the current run of the report writer. This variable is used by BOOMS to display the “PagCount” field on the screen after the report is finished. This variable should not be modified.
- **&PCONTROL** is a value used to determine if “printer setup” is to be done on a report. The default is “Y” (do printer setup). Any other value will cause printer setup to be suppressed. If both **&PCONTROL** and **&HCONTROL** are specified as “N”, the printed output will contain only detail information that would be suitable input to a “spreadsheet”, “word processor” or “database management” program.
- **&PN** is the NEXT page number to be printed. This is reset to 1 by the NEWREPORT Action.
- **&PQUALITY** is the 1-character print quality indicator. “D” is Draft Quality (the default) and “N” is Near Letter Quality.
- **&PRCOUNT** is the number of records printed.
- **&PRINT** refers to the printer number (as seen in the top left corner of most screens). You can use this variable to SET or verify proper printer setup for a report. If you SET this variable, it is reset to its original value after the report has run. **&PRINT** can be set to “T” to print a report to the screen. **&PRINT** can be set to “S” to print a report to the default Spool Drive.
- **&PRINTC** refers to the printer characteristics name (as seen in the top left corner of most screens). You can use this variable to SET or verify proper printer setup for a report. If you SET this variable, it is reset to its original value after the report has run.
- **&RCOUNT** is the number of records read from the Primary File during Record Selection.
- **&REPOS** - used in the RECORDSELECTION SECTION to cause a “delayed repositioning” of the primary input file. Normally, when a REPOSITION action is encountered in the RECORDSELECTION SECTION, repositioning occurs immediately. However, if you want to fully process the current record before repositioning, setup the “reposition” record key as needed and set **&REPOS** to “YES”. This will cause repositioning to occur after the current record has been Selected/Rejected. This variable is automatically turned off (reset to “NO”) each time a new record is passed through the RECORDSELECTION SECTION.
- **&REPROC** - set this variable to “YES” in the RECORDSELECTION SECTION to cause the current record to be reprocessed. This variable is automatically turned off (reset to “NO”) each time a new record is passed through the RECORDSELECTION SECTION. Reprocessing the current record allows you to include it more than once for Sorting/Printing purposes. This is especially useful when processing the Orders file and you want to include an Internal Order twice (once for Management and once for Brokerage).
- **&ROWS** contains the number of displayed rows on the screen
- **&RWVAR1 / &RWVAL1** are the first pair (Level 1) of SORTSUMMARY variables used to sort records based on a cumulative total that is only known after all records have been selected
- **&RWVAR2 / &RWVAL2** are the second pair (Level 2) of SORTSUMMARY variables used to sort records based on a cumulative total that is only known after all records have been selected
- **&SIGNONID** - is the 8-character user Signon-ID name
- **&SORTFLD1... &SORTFLD9** map to the 1st nine SORTDEFINITION fields. These fields are valid only after the SORTDEFINITION SECTION has been processed (i.e. they are intended for use in the PRINT SECTION). If any of them are specified, the Report Writer will retain the entire Sort Output Record (BOOMS.RPO.TMP) and its fields will be available for use in SET statements in the PRINT SECTION. This allows you to associate values with individual records that are not part of the original record (perhaps a variable assigned during record selection).
- **&SPACES** - is a character string of 100 “blanks”
- **&SRCOUNT** is the number of records input to the Sort.
- **&STABSIZ** is the size of the SORTSUMMARY Table. The value ranges from 100 to 4096 (1000 is the default). The default value uses an additional 4K of RAM. The maximum value, 4096, uses an additional 64K of RAM. Only change the value if you are running a complex SORTSUMMARY that requires a large internal sort (i.e. increase the value if you receive error message):

#### **SORTSUMMARY TABLE OVERFLOW ...**

If you try to use more memory than is available on your system, you will receive error message:

#### **INSUFFICIENT MEMORY FOR SORTSUMMARY ...**

- **&STOPMSG** is a 38-character variable which allows you to set the text for the error message to be displayed when a “STOP” action is encountered in the SETUP SECTION (i.e. the message to be displayed in place of

“FORCED STOP DURING REPORT SETUP”)

- **&WDATE1 ... &WDATE6** are screen variable DATES
- **&WNUMBR1 ... &WNUMBR5** are screen variables for numeric values (whole numbers)
- **&WDOLLR1 ... &WDOLLR3** are screen variables for numeric values (dollars and cents)
- **&WCODE1, &WCODE2, &WCODE7 ... &WCODEC** are screen variables for character values (8 characters)
- **&WCODE3, &WCODE4, &WCODE5** are screen variables for character values (30 characters)
- **&WILDCARD** can be set to a “wildcard” character (e.g. &WILDCARD = “\*”) to allow “wildcard” comparisons. For example, if &WCODE1 contains “ABC\*”, the comparison:

```
IF MAILERCODE = &WCODE1
```

will be true if the first 3 characters of MAILERCODE are “ABC”. NOTE: the variable containing the wildcard MUST be to the right of the equals sign

- **&XRAWREC** is a flag that, when set to “Y”, indicates that a database raw record is being written. This flag is required when stripping raw records from any of the BOOMS files.

### &KEYNUM/&KEYSEGN Combinations By File Id (FID)

**FID(AC)**

&KEYNUM = 0	&KEYSEG0	=	“R” or “P”
	&KEYSEG0[1,1]	=	Binary 0
	&KEYSEG0[5,2]	=	account code
	&KEYSEG1	=	payment due date
	&KEYSEG2	=	order number
&KEYNUM = 1	&KEYSEG0	=	“R” or “P”
	&KEYSEG0[1,1]	=	binary 0
	&KEYSEG0[5,2]	=	account code
	&KEYSEG1	=	order number
&KEYNUM = 2	&KEYSEG0	=	“R” or “P”
	&KEYSEG0[1,1]	=	binary 0
	&KEYSEG1	=	order number
	&KEYSEG2	=	account code
&KEYNUM = 3	&KEYSEG0	=	“R” or “P”
	&KEYSEG0[1,1]	=	binary 0
	&KEYSEG0[5,2]	=	account code
	&KEYSEG1	=	client PO number
	&KEYSEG2	=	order number
&KEYNUM = 4	&KEYSEG0	=	“R” or “P”
	&KEYSEG0[1,1]	=	binary 0
	&KEYSEG0[5,2]	=	account code
	&KEYSEG1	=	mailer code
	&KEYSEG2	=	list number
	&KEYSEG3	=	order number
&KEYNUM = 5	&KEYSEG0	=	“R” or “P”
	&KEYSEG0[1,1]	=	binary 0
	&KEYSEG0[5,2]	=	account code
	&KEYSEG1	=	post date
	&KEYSEG2	=	order number

**FID(DA)**

&KEYNUM = 0	&KEYSEG0	=	datacard number
-------------	----------	---	-----------------

**FID(EX)**

&KEYNUM = 0	&KEYSEG0[5,0]	=	mailer code 1
-------------	---------------	---	---------------

	&KEYSEG0[5,5]	=	mailer code 2
	&KEYSEG1	=	order number
&KEYNUM= 1	&KEYSEG0	=	0 (control record)
	&KEYSEG1[5,0]	=	mailer code 1
	&KEYSEG1[5,5]	=	mailer code 2
<b>FID(GL)</b>			
&KEYNUM = 0	&KEYSEG0	=	account number
	&KEYSEG1	=	subaccount number
	&KEYSEG2	=	ledger date
	&KEYSEG3	=	transaction source
&KEYNUM = 1	&KEYSEG0	=	transaction number
	&KEYSEG1	=	account number
	&KEYSEG3	=	ledger date
<b>FID(JO)</b>			
<b>FID(JP)</b>			
<b>FID(JH)</b>			
&KEYNUM = 0	&KEYSEG0	=	journal code
	&KEYSEG1	=	transaction number
&KEYNUM = 1	&KEYSEG0	=	transaction number
	&KEYSEG1	=	journal code
&KEYNUM = 2	&KEYSEG0	=	account number
	&KEYSEG1	=	journal code
	&KEYSEG2	=	transaction number
&KEYNUM = 3	&KEYSEG0	=	order number
	&KEYSEG1	=	account number
	&KEYSEG2	=	transaction number
	&KEYSEG3	=	journal code
&KEYNUM = 4	&KEYSEG0	=	journal code
	&KEYSEG1	=	the account to post
	&KEYSEG2	=	transaction number
<b>FID(LI)</b>			
&KEYNUM = 0	&KEYSEG0	=	list number
&KEYNUM = 1	&KEYSEG0	=	alternate sort code
	&KEYSEG1	=	list number
&KEYNUM = 2	&KEYSEG0	=	owner code
	&KEYSEG1	=	list number
<b>FID(OR)</b>			
&KEYNUM = 0	&KEYSEG0	=	order number
&KEYNUM = 1	&KEYSEG0	=	list number
	&KEYSEG1	=	mailer code
	&KEYSEG2	=	order number
&KEYNUM = 2	&KEYSEG0	=	mailer code
	&KEYSEG1	=	list number
	&KEYSEG2	=	order number
&KEYNUM = 3	&KEYSEG0	=	mailer code
	&KEYSEG1	=	begin mail date
	&KEYSEG2	=	order number
&KEYNUM = 4	&KEYSEG0	=	broker code
	&KEYSEG1	=	mailer code



	&KEYSEG2	=	list number
	&KEYSEG3	=	order number
<b>FID(PR)</b>			
<b>FID(BR)</b>			
<b>FID(OW)</b>			
<b>FID(ML)</b>			
<b>FID(SH)</b>			
&KEYNUM = 0	&KEYSEG0[1,0]	=	type
	&KEYSEG0[5,1]	=	promotion code
&KEYNUM = 1	&KEYSEG0	=	type
	&KEYSEG1	=	alternate sort code

## Valid Arithmetic and Logical Connectors

Following is a list of the valid Arithmetic Connectors for numeric operators:

+	addition
-	subtraction
*	multiplication (not valid for date arithmetic)
/	division (not valid for date arithmetic)
()	operators can be groups according to the rules of arithmetic

Following is a list of the valid Logical Connectors:

<b>OR</b>	logical OR
<b>AND</b>	logical AND
<b>NOT</b>	negative (leading) connector
<b>XOR</b>	logical "exclusive OR"
=	equals
<b>NE</b>	not equal
<>	not equal
<b>GT</b>	greater than
>	greater than
<b>LT</b>	less than
<	less than
<b>GTE</b>	greater than or equal
>=	greater than or equal
<b>LTE</b>	less than or equal
<=	less than or equal

**@IN** in a statement of the form, **string\_1 @IN string\_2**, @IN does a substring search for "string\_1" in "string\_2". It returns a "True" (YES) condition if it is found. For example, if the variable, &ZIP, contains the value 'ZIP' and SELECTION1 contains the value '\*\*\*zip select', then the expression "**IF &ZIP @IN SELECTION1**" is true.

**@NOTIN** is the opposite of @IN

**@INF** in a statement of the form, **string\_1 @INF string\_2**, @INF does a **fixed position** search for "string\_1" in "string\_2". It returns a "True" (YES) condition if it is found. The fixed positions in "string\_2" depend on the length of "string\_1". So, if "string\_1" has a length of 2, then the 2 characters in "string\_1" are compared to the 2 characters in "string\_2" at positions 0,2,4,6,8 and so on, until the end of "string\_2". For example, if the variable &CATCOD contains the value 'AB' and CATEGORYCODS contains the value 'AABBAB', then the expression "**IF &CATCOD @INF CATEGORYCODS**" is true. However, if CATEGORYCODS contains 'AABBCC' then the expression would be false (but the expression "**IF &CATCOD @IN**

**CATEGORYCODS**” would be true since this search would find a match at position 1).

**@@NOTINF** is the opposite of **@INF**

## Report Writer Menu

A Report Writer Menu is a screen that you can tailor to look similar to one of the menu screens from the built in reporting system (e.g. the Order Reporting Screen (“O2”) from the Main Selection Menu). This allows you to access R/W reports without having to know the underlying “DOS” names of the Report Description Files. You can build a customized menu for all signon-IDs or customize a different menu for each signon-ID (The name of the menu is placed in a user’s Signon-ID record) . The concept is the same as the built-in menus: a 2-character code is associated with a report to be run. Menu files (.LMU files) are text files (like .LRL files) that contain entries which are a combination of descriptive text and “LRL” assignments. A “LRL” assignment statement is of the form:

```
{lrlname}XX description
```

where:

“braces” [ {} ] are required and indicate an assignment  
 “lrlname” is the 8 or 12-character **Report Description File name** (blank padded, if necessary). If the extension is not entered, “.LRL” is assumed.  
 “XX” is the 2-character **Report ID (RPTID)** to be associated with “lrlname”.  
 “description” is text to appear on the screen to describe the report.

Each line on the menu can be up to 76 characters in length. The “braces” and “lrlname” do not appear on the screen, so characters to the right are shifted 10 (or 14) characters to the left (i.e. as if the “braces” and “lrlname” were never entered). The “BOOMS REPORTS” distribution disk contains several files (“**BMENU**”, “**BACCT**”, “**BBROK**”, “**BMANA**” and “**BLIST**”) that can be used as prototypes to assist you in customizing your own Menu. The menu is displayed by pressing “RptHelp” (Shift-F1) while the cursor is in the top part of the screen. While on the Help screen, you can enter the 2-character **RPTID** to activate a report or press “F9” to return to the main screen without activating any report.

For example, the following statements were used to create the sample menu, “**BMENU.LMU**” (Figure 27.5):

```

{      }          *-----*
{      }          ***** Sample Report Writer Menu *****
{      }          *-----*
{ORDRPRT5 }01 - List Rental Billing Summary      {ACCTPRT2 }A1 - A/R-A/P Summary (by Account
{ORDRPRT8 }02 - List Usage by Mailer            {ACCTPRT3 }A2 - A/P Outstanding
{ORDRPRTB }04 - Brokerage Sales Report          {ACCTPRT4 }A3 - Credit History (by Mailer)
{ORDRPRT E }05 - Mailer Usage (by List)        {ACCTPRTB }A4 - Credit History (by Broker)
{ORDRPRTI }06 - Managed List Activity Summary  {ACCTPRT5 }A5 - Closed Order Report
{ORDRPRTJ }07 - Broker Usage Summary          {ACCTPRT7 }A6 - Management Accounts Payable
{ORDRPRTQ }08 - Management Sales Summary      {ACCTPRT6 }A8 - Accounting by Order Number

{      }          {JOURPRT1 }J1 - "CONTRAs" Report
{PROMPRT2 }P1 - Promotion Conflicts           {JOURPRT4 }J2 - "FREED" Journal Transactions
{PROMPRT3 }P2 - "DEFERRED" Delete Requests    {JOURPRT5 }J3 - "PIF'D" Journal Transactions
{      }          {JOURPRTB }J4 - Daily Cash Flow
{LISTPRT9 }L1 - List File Printout
{LISTPRT5 }L2 - "DEFERRED" DELETE REQUEST     {GENLPRT1 }G1 - G/L Listing by Sub-Account
{      }          {GENLPRT2 }G2 - G/L Salesperson Expenses/Comms
{DATAPRT1 }D1 - DataCard File Listing         {GENLPRT3 }G3 - G/L Locate REF# (CHECK#)

```

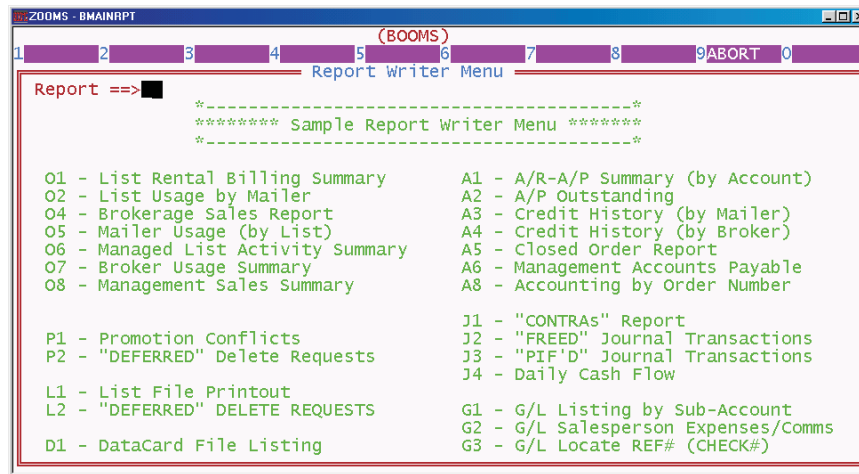


Figure 27.5. Sample Report Writer Menu

## Summary Rankings

The Report Writer has the ability to do a two-level “summary sort” that allows records to be sorted based on a “total” value rather than a value from the record itself. For example, if all you needed to do was sort “Orders” based on “list rental quantity”, you could put the following statement in the SORTDEFINITION SECTION:

```
RENT-QTY,,A
```

However, if you wanted to sort “orders” by “mailer” based on the “total List Rental Quantity” where all Orders for a Mailer with the highest total quantity would appear first, the Mailer with the second highest total would appear second and so forth, you could not use a standard calculation in the RECORDSELECTION SECTION to SET a variable for sorting purposes since the “total quantity” for a particular mailer is not known when a record is selected.

Instead, a special technique is required to accomplish a “summary sort”. Two pairs of “special-meaning” built-in variables are defined for this purpose. Setting these variables is only meaningful in the **RECORDSELECTION SECTION**. The variables are:

```
&RWVAR1 / &RWVAL1
&RWVAR2 / &RWVAL2
```

**&RWVAR1** defines a 10-character field for which a “summary value” is to be maintained. **&RWVAL1** is a value (number) to be added to the total for &RWVAR1 each time a record is selected. For example, if you wanted to maintain “total rental quantity” for a “List”, you could issue the following SET commands:

```
SET &RWVAR1 = LISTNUMBER
SET &RWVAL1 = RENTQTY-REQ
```

The Report Writer will maintain this information in a “SORTSUMMARY” Table (the size of which is determined by a screen variable). After all records have been selected (and prior to the SORT), the Record Selection output records will be updated to reflect the summary value for each record. In order to associate the summary value to a particular sorting sequence, an entry must be placed in the **SORTDEFINITION SECTION** as follows:

```
&RWVAL1,,D
```

There is no restriction on where in the SORTDEFINITION that this line must appear. However, for summary purposes, it is likely to be the first entry.

Following is a more detailed example. Six Orders are to be sorted by List based on total quantity for each list:

Order#	List#	Rental Quantity
1	1000	10,000

2	2000	5,000
3	2000	10,000
4	1000	5,000
5	3000	20,000
6	1000	10,000

By using the above statements in the RECORDSELECTION and SORTDEFINITION SECTIONs you cause the Report Writer to build the SORT SUMMARY Table. After all records have been selected, the SORTSUMMARY TABLE would look as follows (your report definition does not need to know the structure of this table. It is shown here to help clarify the summary process):

List# (&RWVAR1)	TotQTY (&RWVAL1)	Comment
1000	25,000	Total of Orders 1,4,6
2000	15,000	Total of Orders 2,3
3000	20,000	Total of Order 5

Since &RWVAL1 is a variable in the SORTDEFINITION SECTION, the Record Selection Output File is reprocessed. Depending on the List Number, each record will receive an &RWVAL1. Thus, records 1,4 and 6 will get a value of 25000, records 2 and 3 will get a value of 15000 and record 5 will get a value of 20000. After completion of the sort, the records will be in the following sequence:

Order#	List#	Rental Quantity
1	1000	10,000
4	1000	5,000
6	1000	20,000
5	3000	20,000
2	2000	5,000
3	2000	10,000

Thus, records will be input into the PRINT SECTION in the desired ranking sequence. If "RENT-QTY" were added as a second entry in the SORTDEFINITION SECTION, the record would be further sorted by rental quantity within each List.

The variables &RWVAR2/&RWVAL2 are used in a similar manner to summarize at a second level. For example, suppose you want to summarize mailer activity by rental quantity within summarized list activity. In this case, &RWVAR1/&RWVAL1 are used to group list activity. To group mailer activity within list activity, the following statement could be added to the RECORDSELECTION SECTION:

```
SET &RWVAR1 = LISTNUMBER
SET &RWVAR2[5,0] = LISTNUMBER
SET &RWVAR2[5,5] = MAILERCODE
SET &RWVAL1 = RENTQTY-REQ
SET &RWVAL2 = RENTQTY-REQ
```

You would add the following statements to the SORTDEFINITION SECTION:

```
&RWVAL1,,D
&RWVAL2,,D
```

 **For a two-level summary, the variable &RWVAR2 must contain information on both the first and second level sort fields. If information were just given for mailer (i.e. &RWVAR2 = MAILERCODE), the Report Writer would not be able to relate the mailer to a particular list.**

## Include/HInclude Files

The Report Writer contains an “**INCLUDE**” directive capability. This allows an external file to be **embedded** into a report description file (.LRL File) Included files are actually Report Description fragments whose statements function just like statements in the Description File. Embedding is accomplished by placing a statement of the following form at the desired location:

```
#INCLUDE name.LRI
```

The following rules apply:

- The #include statement must begin in column 1 and a “blank” must follow the word “INCLUDE”.
- “name” can be any valid 8.3 “DOS” name. For compatibility, it should be of type “LRI” (Lissan Reporting Include file).
- Nesting is not allowed. That is, you cannot use an “#INCLUDE” directive in an “LRI” file. If you do, it will be ignored.
- There is no limit to the number of “#INCLUDE” directives in the “LRL” file.
- If a file is included that does not exist, it will be ignored.

You may find “INCLUDE” files useful if you are writing several reports that use portions of logic that are the same or if you are writing many different reports that require slightly different logic.

The Report Writer contains an “**HINCLUDE**” directive capability. This allows an external file to be imbedded in a report header that contains a tab-delimited column header line (the statement should be the first statement in the HEADER SECTION):

```
#HINCLUDE name.TXT
```

The following rules apply:

- The #hinclude statement must begin in column 1 and a “blank” must follow the word “HINCLUDE”
- “name” can be any valid 8.3 “DOS” name. For compatibility, it should be of type “TXT”
- Only 1 #HINCLUDE is allowed
- If a file is included that does not exist, it will be ignored.
- This file is included ONLY when “Force TABS/SPOOL” is set to “Y”.