Manually Entering or Editing Batch Report Parameters

This section describes parameter syntax and statements to be used if you are entering the batch parameter statements manually or are making changes to statements generated by the GENCARD command.

This chapter covers the following topics:

- Parameter Statement Syntax
- Parameter Statements
- Parameter Statement Processing

Parameter Statement Syntax

A parameter statement consists of a statement name followed by at least one blank and one or more optional (positional or keyword) parameters separated by commas:

```
statement name    parm1, parm2, parm3 ,...
```

The following syntax rules pertain to parameter statements:

- Statements may be entered in positions 1-71.
- At least one blank must separate the statement name and the first parameter.
- Multiple parameters must be separated by commas.
- Blanks are not permitted within a parameter entry, except when enclosed within apostrophes.
- A statement with multiple parameters may be continued by ending the line with a comma followed by a blank, and entering the next parameter on the next line beginning in any position between 1-71.
- Comment lines may be inserted by entering an asterisk in position 1. Comment lines may also be inserted following the blank that ends the statement.

Positional Parameters

Positional parameters specify the data fields to be summarized and/or displayed as part of the Adabas Review report.

A positional parameter is a single value, usually an Adabas command log field name. Refer to section *Field Reference* for information about individual data fields.

Keyword Parameters

Keyword parameters either specify input record selection criteria or define specific parameter statement options. A keyword parameter consists of a keyword name, a keyword separator, and a parameter value.

Keyword names vary depending on the particular parameter statement. Field names referencing Adabas data fields are used as keyword names when specifying input record selection criteria. When defining a specific parameter statement option, specific keyword reserved words are used as keyword names.

Valid keyword separators include:

```
=, <,>, and ¬
```

These relational symbols (equal to, less than, greater than, and not equal to, respectively) are used to define selection criteria for Adabas Review report processing.

Parameter values are numeric, alphanumeric, or hexadecimal in format. Note that a hexadecimal value must be enclosed in apostrophes and be preceded by an "X". A list of values is identified by enclosing the list in parentheses with the values separated by commas. For a range of values, a hyphen is used to separate the low and high values of the range.

Parameter Statements

A summary table of the parameter statements used by the Adabas Review processor is presented below. For information about the data fields used with the statements, refer to section *Field Reference*.

Statement	Use
AVERAGE	Specifies the data fields for calculating average values in summary reports.
COPY	Copies input detail records or summary data records to a sequential data set or the Adabas Review repository.
COST	Specifies the factors to be used in resource cost calculations for summary records.
DISPLAY	Defines the control break fields for summary reports, and specifies the data fields to be printed for detail reports.
ENV	Specifies a numerical factor for adjusting values returned in the CMDRESP field.
ENVIRONMENT	Allows the user to specify CPU-ID and database ID. This information is included in output generated by a COPY statement for a report.
EXCLUDE	Defines specific input detail records to be excluded from processing for both summary and detail reports.
INCLUDE	Defines specific input detail records to be included in processing for both summary and detail reports.
INPUT	Defines the input data to be processed by the Adabas Review processor and optional output logging parameters.
LOG	Specifies how Adabas Review performs physical command logging.
MINIMUM	Specifies the data fields for which minimum values will be printed on summary reports.
PERCENT	Specifies the data fields for which percentage is to be calculated for summary reports.
RATE	Specifies the data fields for which rate is to be calculated for summary reports.
REPORT	Defines the type of report to be generated along with the report format and title lines.
ROUND	Specifies the data fields for which rounding is to occur.
SUM	Specifies the data fields for which total values will be produced on summary reports.

AVERAGE Statement

AVERAGE *field-name*, *field-name*,...

The AVERAGE statement specifies the data fields for which average values are to be calculated for an Adabas Review summary report.

The AVERAGE statement is applicable to summary reports only. Average values are calculated only for the valid data fields specified in this statement. Average values are summarized and printed on the summary report at each control break defined by the parameter statement for all data fields for which average values have been calculated. Average values are printed to one more rounded decimal place than the field data being averaged.

Example:

Print average values for the Adabas command log fields ASSOIO, WORKIO, and DATAIO on the summary report.

```
AVERAGE ASSOIO,WORKIO,DATAIO
```

COPY Statement

```
COPY  FILETYPE = {SEQUENTIAL | ADABAS}
      LIMIT = numeric-value
      DBID = database-id
      FILE = file-number
      SVC = svc
```

where DBID, FILE, and SVC represent the Adabas Review repository.

The COPY statement specifies that input detail records or summary data records are to be copied to a sequential data set or an Adabas file.

The keyword parameter FILETYPE specifies whether the data is to be copied to a sequential data set or to Adabas. The following functions can be performed using this statement:

- **Copy detail records to a sequential data set.**
  The user can select subsets of detail log records and copy them to different sequential data sets. These data sets can then be used as future input to Adabas Review.

- **Copy detailed or summary data to Adabas.**
  The user can write specified information back into the Adabas Review repository where it can be viewed using either the online history functions or Adabas applications written by the user.

The physical command log may be created with less overhead for the Adabas nucleus by using this function with the Adabas Review processor running interactively.

The keyword parameter LIMIT specifies the maximum number of records to be copied. The default is to copy all records.

Examples:

1. Copy all input detail records satisfying the selection criteria specified on the INCLUDE statement to the sequential data set RVUCOPnn, where nn is incremented sequentially beginning with 01 for the first copied file.

```
REPORT TYPE=DETAIL
INCLUDE  ...
DISPLAY  ...
COPY FILETYPE=SEQUENTIAL
```

2. Copy all summary data records to the Adabas Review repository.

```
REPORT TYPE=SUMMARY
DISPLAY  ...
COPY FILETYPE=ADABAS,DBID=133,FILE=3
```

COST Statement

COST {*field1-name arithmetic-operator cost-factor, ...* | **FIXED** }

The COST statement, which is used only in summary reports, specifies the factors to be used in calculating resource costs and the data fields to which the cost factors are to be applied. The costs calculated for the specified data fields are automatically printed on the output report. If costs are calculated for more than one data field, a combined total cost column is printed as the column to the far right on the summary report. The special reserved word FIXED specifies that a fixed cost is to be applied to each input detail record.

Valid arithmetic operators are: *, +, -, and @. The first three operators specify that the value of the designated data field is to be multiplied by the indicated cost factor (*), that a constant factor is to be added to the value of the data field (+), or that a constant value is to be subtracted from the value of the data field (-). The remaining operator (@) is used only with FIXED to specify that a fixed cost is to be applied to each input detail record.

Cost factors can be positive or negative numeric values containing up to four decimal places. Negative values can be used only with the operators * and @. Costs are calculated only for those data fields for which cost factors are specified.

Examples:

1. Calculate costs by multiplying the CMDRESP field by .0025 and multiplying the total IO count by 2.50.

```
COST CMDRESP*.0025,IO*2.50
```

2. Calculate costs by applying a fixed cost of 5.00 to each input detail record.

```
COST FIXED@5.00
```

3. Calculate costs by adding 1.5 to the value of the ASSOIO field, multiplying the DATAIO field by a negative 1.25, multiplying the WORKIO field by a positive 2.

```
COST ASSOIO+1.5,DATAIO*-1.25,WORKIO*
```

DISPLAY Statement

DISPLAY *field-name,field-name,...*

The DISPLAY statement defines the control break fields for Review summary reports. Data is summarized at each control break specified in this statement for all data fields for which calculations have been performed.

One DISPLAY statement *must* be specified with at least one field for a summary report.

The DISPLAY statement is also used to specify the data fields to be printed on Review detail reports.

Examples:

1. Print data summarized by Adabas COMMAND within file number (FNR) within JOBNAME on the summary report.

```
DISPLAY JOBNAME,FNR,COMMAND
```

2. Print the data fields SEQUENCE, DATE, USERID, and CMDRESP on the detail report in the order specified.

```
DISPLAY SEQUENCE,DATE,USERID,CMDRESP
```

ENV Statement

ENV ENV-FACTOR = *nnnnn.nnnn*

Note:
Adabas Review makes no claim to be able to recover or report actual CPU time.

If you are not satisfied with the values you receive in the Adabas Review CMDRESP field, you can use the ENV control statement to adjust the values.

"nnnnn.nnnn" is the factor to be multiplied by the value reported in the Adabas Review field CMDRESP to adjust it to the total Adabas nucleus session CPU time reported by the operating system. All digits must be specified. The default value is 1.0000; the maximum value is 99999.9999.

To implement the ENV control statement

1. Run your Adabas nucleus with an autostarted Review report that shows the total of all command response time for the nucleus session (CMDRESP field).
2. When the database is stopped, examine the output report and note the total CMDRESP value as reported by Adabas Review.
3. Obtain the total CPU time used by the Adabas nucleus session as reported by the operating system.
4. Compare the two values and determine the relationship between them.
5. For the next Adabas nucleus session, you can adjust the Adabas Review value to more closely approximate the operating system value by inserting the ENV control statement into the RVUAUT1/2 or RVUPARM after the INPUT statement and before the first REPORT statement.

Examples:

1. Review CMDRESP=100.0000; CPU time reported by the operating system=50.0000.

   The ratio between the two values is 2:1 where the Adabas Review CMDRESP field value is two times the Adabas nucleus session CPU time reported by the operating system.

   To adjust this for the next Adabas session, set the ENV control statement parameter ENV-FACTOR to 0.5000 so that Review divides the CMDRESP value by 2.

   The RVUAUT statements may be similar to the following:

```
INPUT FILETYPE=SEQUENTIAL,
      BUFFER-SEGMENTS=2000,
      FILES=1,
      REVIEW-COMMANDS=YES
ENV ENV-FACTOR=0.5000
REPORT TYPE=SUMMARY,
       TITLE='TOTAL CMDRESP',
       PROGRAM=RD-00089,
       RESTRART=Y,
       MAXSTORE=8
       TARGET=00nnn
DISPLAY DBNAME
SUM COMMANDS,IOS,CMDRESP
```

2. Review CMDRESP=25.0000; CPU time reported by the operating system=100.0000.

   The ratio between the two values is 1:4 where the Adabas Review CMDRESP field value is one quarter of the Adabas nucleus session CPU time reported by the operating system.

   To adjust this for the next Adabas session, set the ENV control statement parameter `ENV-FACTOR` to 4.0000 so that Review multiplies the CMDRESP value by 4.

ENVIRONMENT Statement

```
ENVIRONMENT { CPU-ID = cpu-id | DATABASE-ID = database-id }
```

The ENVIRONMENT statement specifies the identification number of the CPU or database from which the Adabas command log originated. The ENVIRONMENT statement should only be used with the COPY statement (see *COPY Statement* for more information).

EXCLUDE Statement

```
EXCLUDE field-name relation-symbol (value, value,...), ...
```

The EXCLUDE parameter statement defines specific input detail records that are to be excluded from processing.

Relational symbols specify that input detail records are to be excluded if the value of the designated data field is equal to, less than, greater than, or not equal to the value indicated in the parameter statement:

	Symbols	Hex Equivalent
Equal to	=	X’7E’
Less than	<	X’4C’
Greater than	>	X’6E’
Not equal to	¬	X’5F’

Note:
The symbol for "not equal to" varies; it may also be "^" or "". The hex equivalent is the same on all machines.

Parameter values may be in numeric, alphanumeric, or hexadecimal format. A hexadecimal value must be enclosed in apostrophes and preceded by an "X". A list of values is enclosed in parentheses with the values separated by commas. For a range of values, the low and high values of the range are separated by a hyphen.

For alphanumeric fields, you may use an asterisk (*) for "wildcard" prefixing and suffixing. For example, to exclude all job names beginning with "A" and all job names ending with "TEST", enter the statement as

```
EXCLUDE  JOBNAME=(A*,*TEST)
```

Note:
EXCLUDE parameter statements with multiple field names have different logical meanings if the field names are entered as multiple EXCLUDE statements rather than as a single continuous statement. If entered as a single statement, the Boolean operator AND is implied; if entered as multiple statements, the Boolean operator OR is implied.

Examples:

1. Exclude input detail records from processing if the value of the SEQUENCE field is less than 100 AND the value of the JOBNAME field is equal to AAAAAAAA or is within the range XXXXXXXX through ZZZZZZZZ.

```
EXCLUDE
SEQUENCE<100,JOBNAME=(AAAAAAAA,XXXXXXXX-ZZZZZZZZ)
```

2. Exclude input detail records from processing if the value of the SEQUENCE field is less than 100 OR the value of the JOBNAME field is equal to AAAAAAAA or is within the range XXXXXXXX through ZZZZZZZZ.

```
EXCLUDE SEQUENCE<100
EXCLUDE JOBNAME=(AAAAAAAA,XXXXXXXX-ZZZZZZZZ)
```

INCLUDE Statement

INCLUDE *field-name relation-symbol* (*value, value,...*), ...

The INCLUDE parameter statement defines specific input detail records to be included in processing.

Relational symbols specify that input detail records are to be included if the value of the designated data field is equal to, less than, greater than, or not equal to the value indicated in the parameter statement.

	Symbols	**Hex Equivalent**
Equal to	=	X’7E’
Less than	<	X’4C’
Greater than	>	X’6E’
Not equal to	&#65533;	X’5F’

Note:
The symbol for "not equal to" varies; it may also be "^" or "". The hexadecimal equivalent is the same on all machines.

Parameter values may be in numeric, alphanumeric, or hexadecimal format. A hexadecimal value must be enclosed in apostrophes and must be preceded with an "X". A list of values is enclosed in parentheses with the values separated by commas. For a range of values, the low and high values of the range are separated by a hyphen.

For alphanumeric fields, you may use an asterisk (*) for "wildcard" prefixing and suffixing. For example, to include all job names beginning with "A" and all job names ending with "TEST", enter the statement as

```
INCLUDE  JOBNAME=(A*,*TEST)
```

Note:
INCLUDE parameter statements with multiple field names have different logical meanings if the field names are entered as multiple INCLUDE statements rather than as a single continuous statement. If entered as a single statement, the Boolean operator AND is implied; if entered as multiple statements, the Boolean operator OR is implied.

An individual field name may only be specified once in an INCLUDE statement. For example, the following INCLUDE statement is invalid because it specifies the SEQUENCE field twice:

```
INCLUDE  SEQUENCE>100,SEQUENCE=3
```

Examples:

1. Include input detail records in Adabas Review processing only if the value of the SEQUENCE field is greater than 100, AND the value of the JOBNAME field is equal to AAAAAAAA or ZZZZZZZZ, AND the value of the response code field (RSP) is within the range 9 through 150.

   Note:
   The same field name may not be used more than once for a particular INCLUDE statement.

   ```
   INCLUDE
   SEQUENCE>100,JOBNAME=(AAAAAAAA,ZZZZZZZZ),RSP=(9-150)
   ```

2. Include input detail records in Adabas Review processing if the value of the SEQUENCE field is greater than 100, OR if the value of the JOBNAME field is equal to AAAAAAAA or ZZZZZZZZ, OR if the value of the response code (RSP) field is within the range 9 through 150.

   ```
   INCLUDE  SEQUENCE>100
   INCLUDE  JOBNAME=(AAAAAAAA,ZZZZZZZZ)
   INCLUDE  RSP=(9-150)
   ```

INPUT Statement

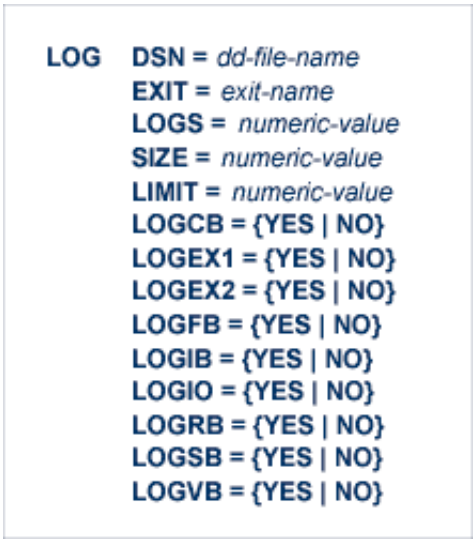
```
INPUT   {FILETYPE | FILE-TYPE} = SEQUENTIAL
        {LIMIT | LOG-LIMIT} = numeric-value
        BUFFER-SEGMENTS = numeric-value
        FILES = numeric-value
        REVIEW-COMMANDS = {YES | NO}
        SKIP = numeric-value
```

—where FILES is used only under z/VSE.

The INPUT parameter statement defines the characteristics of the input records to be processed by the Adabas Review processor. The statement format is described in the following table:

Keyword	Specifies . . .
FILETYPE	the file type of the input records to be processed. The default and the only value currently possible is SEQUENTIAL, which implies Adabas command log record input or input from the interactive processor.
LIMIT	the maximum number of input records to be processed. The default is to process all of the input records. A value of "99999999" indicates that there is no limit.
BUFFER-SEGMENTS	the number of 128-byte segments to be obtained for the Review buffer pool when running the Adabas Review interactive processor. The default value is 700 or approximately 87K for z/VSE, 400 (or 50K) for z/VM or BS2000, or 2000 (or 250K) for z/OS.
FILES	(z/VSE only) the number of input command log files to be processed.
REVIEW-COMMANDS	whether commands issued by Review should be included in the command processing for all reports. REVIEW-COMMANDS=NO indicates that special Adabas commands with ACBFNR=0 and ACBRSP=17 and special Review 4.3 commands are skipped for Review communication purposes; REVIEW-COMMANDS=YES indicates that those commands are also displayed.
SKIP	the number of command log records to bypass before Adabas Review begins processing.
REVLOGMAX	When the sum of sizes of the logged buffers for one command on the Adabas command log reaches the value of the REVLOGMAX parameter, the buffer exceeding the limit is truncated and all following buffers are omitted for internal processing. The minimum setting is 2000; the maximum setting is 32768, and this is also the default. If buffers are omitted, fields may not be filled.
REVLOGBMAX	If a buffer is longer than this value, the logged buffer is truncated from the point at which its size exceeds the setting of the REVLOGBMAX parameter. The minimum setting is 0; the maximum setting is 30000, and the default value is 1024. If buffers are truncated, fields or parts of fields may not be filled.

LOG Statement



The LOG statement determines how Adabas Review is to perform physical command logging; i.e., what information is to be logged and where it is to be written.

The following table describes the parameters within the LOG statement:

Parameter	Specifies . . .
DSN	the file name prefix for the file where the command logs are to be written. A number is appended to this name based on the LOGS parameter. If DSN=RVLOG and LOGS=2, the command log data is written to files RVLOG01 and RVLOG02.
EXIT	the name of the user exit to be called when a command log data set is closed and before the next command log data set is opened.
LOGS	the number of command log data sets for the report.
SIZE	the maximum number of blocks allocated to the command log data set. When the SIZE parameter value is reached, the exit specified in the EXIT parameter is called, and the next command log data set is opened for output.
LIMIT	the maximum number of detail lines to be logged. The default is to log all detail lines. A value of "99999999" indicates that there is no limit.
LOGCB	whether the Adabas control block is to be logged.
LOGEX1	whether the Review command log extension part 1 is to be logged.
LOGEX2	whether the Review command log extension part 2 is to be logged.
LOGFB	whether the format buffer is to be logged.
LOGIB	whether the ISN buffer is to be logged.
LOGIO	whether I/O information is to be logged.
LOGRB	whether the record buffer is to be logged.
LOGSB	whether the search buffer is to be logged.
LOGVB	whether the value buffer is to be logged.

MAXIMUM Statement

The MAXIMUM parameter statement applies only to Adabas Review summary reports. It specifies the data fields for which maximum values are to be printed on these reports. Maximum values are printed only for the valid data fields specified in this statement.

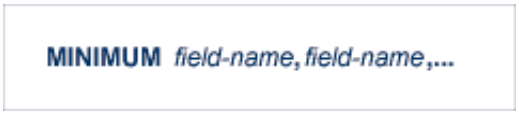
Maximum values are summarized and printed on the summary report at each control break defined by the DISPLAY parameter statement for all data fields for which maximum values have been determined.

Example:

Determine maximum values for the Adabas log fields ASSOIO, WORKIO, and DATAIO and print them on the summary report.

```
MAXIMUM  ASSOIO,WORKIO,DATAIO
```

MINIMUM Statement



The MINIMUM parameter statement applies only to Adabas Review summary reports. It specifies the data fields for which minimum values are to be printed on these reports. Minimum values are printed only for the valid data fields specified in this statement.

Minimum values are summarized and printed on the summary report at each control break defined by the DISPLAY parameter statement for all data fields for which minimum values have been determined.

Example:

Determine minimum values for the Adabas log fields ASSOIO, WORKIO, and DATAIO and print them on the summary report.

```
MINIMUM  ASSOIO,WORKIO,DATAIO
```

PERCENT Statement

The PERCENT parameter statement applies only to Adabas Review summary reports. It specifies the data fields for which percentage values are to be printed on these reports. Percentage values are printed only for the valid data fields specified in this statement.

Percentage values are summarized and printed on the summary report at each control break defined by the DISPLAY parameter statement for all data fields for which percentage values have been determined.

Example:

Determine percentage values for the Adabas log fields ASSOIO, WORKIO, and DATAIO and print them on the summary report. The sum of the percentage values for each specified data field is 100 percent.

```
PERCENT  ASSOIO,WORKIO,DATAIO
```

RATE Statement

The RATE parameter statement applies only to Adabas Review summary reports. It specifies the data fields for which rate values are to be printed on these reports. Rate values are printed only for the valid data fields specified in this statement.

Rate values are summarized and printed on the summary report at each control break defined by the DISPLAY parameter statement for all data fields for which rate values have been determined.

Example:

Determine rate values for the Adabas log fields ASSOIO, WORKIO, and DATAIO and print them on the summary report. The rate will be calculated as the amount of data collected per second.

```
RATE  ASSOIO,WORKIO,DATAIO
```

REPORT Statement

The REPORT statement is critical in that it indicates that all of the following statements up to the next REPORT statement define the contents of the current report.

```
REPORT  TYPE = { DETAIL | SUMMARY }
        [ ADALIMIT = { count | 0 } ]
        [ BREAK = { YES | NO } ]
        [ DISPLAYBY = { SORTED | SUMFIELD | USAGE | SORTEDDE | DATETIME } ]
        [ ENTRIES = { count | 999999 } ]
        [ HISTORY-INTERVAL = { minutes | 0 } ]
        [ LIMIT = { count | 99999999 } ]
        [ {LINESIZE | WIDTH} = { count | 133 } ]
        [ MAXSTORE = { kilobytes | 8 } ]
        [ {PAGESIZE | PAGE-SIZE} = { count | 55 } ]
        [ PRINT = { YES | NO } ]
        [ PROGRAM = { pgm-name | RD-DEFLT } ]
        [ REPORT-EXIT = detail-rpt-exit-name ]
        [ RESTART = { YES | NO }
        [ {SKIP | SKIPPING} = { 1 | 2 } ]
        [ {SPACE | SPACING} = { count | 1 } ]
        [ SUMMARY-EXIT = summary-rpt-exit-name ]
        [ TARGET = database-id ]
        [ {TITLE = " character-string " | TITLE1 = " character-string " TITLE2 = " character-string "} ]
        [ WRAP = {YES | NO } ]
```

The REPORT statement defines the type of report to be generated by the Adabas Review processor. The following table describes the parameters in the REPORT statement:

Keyword	Specifies . . .	Default
TYPE	whether the report to be generated is a detail or summary report. Valid values are DETAIL (generate a detail report) or SUMMARY (generate a summary report). This parameter must be specified; there is no default.	---
ADALIMIT (Summary report only)	a minimum command count (number of times the command was processed) for printinga summary report. For example, if ADALIMIT=100, only entries with a command count of 100 or higher are printed.	0

Keyword	Specifies . . .	Default
BREAK	whether subtotals are printed at control breaks or suppressed. Valid values are YES (print subtotals) and NO (do not print subtotals).	YES
DISPLAYBY	the order in which the data is displayed. Valid values are: **SORTED** Display data in ascending order by control break. **SUMFIELD** Display data in descending order by the first field marked as a sum field. **USAGE** Display data in descending order by usage (e.g., most used to least used command). **SORTEDDE** Display data in descending order by control break. **DATETIME** Display data in ascending order by the start date and time of the control break. interval.	SORTED
ENTRIES	the maximum number of entries that a report will process, thus restricting the amount of data collected by the summary report.	999999
HISTORY-INTERVAL Summary report only	the frequency (in minutes) with which history data is written for a summary report.	0
LIMIT	the maximum number of detail lines to be printed on the output report. The default is to print all detail lines. A value of "99999999" indicates that there is no limit.	99999999
{LINESIZE | WIDTH}	the width of a report line. The line width can be stated as any numeric value of at least 72 characters and not greater than 989 characters.	133
MAXSTORE	the maximum amount of storage (in kilobytes) available for the report.	8
PAGESIZE	the length (in lines) of a report page. The minimum is 10 lines. The default is 55 lines, which provides a top and bottom margin for standard printer spacing on a total page size of 66 lines.	55
PRINT	whether the report will be printed at database termination. Valid values are YES (print the report at database termination) and NO (do not print the report at database termination).	YES

Keyword	Specifies . . .	Default
PROGRAM	the name of the display program to be used if the report results are displayed online.	RD-DEFLT
REPORT-EXIT	the name of the report user exit that is executed whenever a command log record is selected for a detail report. See the section *Report Exits* in *Report Option Parameters* for more information. There is no default for this parameter.	no exit is called
RESTART	whether the report is reactivated after the history interval is reached, the report is closed, or the report MAXSTORE limit is exceeded. Valid values are YES (reactivate the report) and NO (do not reactivate the report).	NO
SKIP	whether to single-space or double-space the detail lines on the output report. The default is to single-space the detail lines. Valid values are 1 (single-space the output report) and 2 (double-space the output report).	1
SPACE	the number of spaces to allow between the data fields printed on the output report. This factor applies to all data fields on detail reports but only to the fields defined as control breaks by the DISPLAY parameter statement on summary reports. The default is to allow one space between data fields.	1
SUMMARY-EXIT	the name of the report user exit that is executed whenever an Adabas command or a summary record is selected for a summary report. See the section *Report Exits* in *Report Option Parameters* for more information. There is no default for this parameter.	no exit is called
TARGET	the database from which the report collects data. If this parameter is missing, invalid, or zero, the report is ignored and will not be started.	0
TITLE	the title line for the report when only one line is used. A maximum of 60 characters is allowed. The default value is blanks.	blanks
TITLE1 TITLE2	the title lines for the report when more than one line is used. A maximum of 60 characters is allowed for each title line. The default value is blanks.	blanks
WRAP	whether wrapping (i.e., reusing data elements for the report) will occur. Valid values are YES (wrapping should occur) and NO (wrapping should not occur).	NO

Example:

Define a summary report with the title "A Report with Exit" to collect data from database 12345. The display program RD-00001 is to be used if the report results are displayed online. The maximum storage limit for the report will be 16 kilobytes and the report will be automatically reactivated when the maximum storage limit is exceeded. The user-written exit MYEXIT will be called whenever a command log record is selected for the report.

```
REPORT  TYPE=SUMMARY,TITLE='A REPORT WITH
EXIT',
PROGRAM=RD-00001,TARGET=12345,RESTART=Y,
REPORT-EXIT=MYEXIT,MAXSTORE=16
```

See the section *Parameter Statement Processing* for another example.

ROUND Statement

The ROUND parameter statement applies only to Adabas Review summary reports. It specifies the data fields for which rounding is to occur on these reports. The fields specified on the ROUND statement must also be specified on the DISPLAY statement.

Example:

Round the Adabas DURATION field up to .05 of a second.

```
ROUND DURATION=.05
```

SUM Statement

The SUM parameter statement applies only to Adabas Review summary reports. It specifies the data fields for which value totals are to be calculated on these reports. Values are printed on the summary report at each control break defined by the DISPLAY parameter statement for all data fields that have been summed.

Example:

Print the summed values for the DURATION field, Associator IOs, Data Storage IOs, and Work IOs on the summary report.

```
SUM     DURATION,ASSOIO,DATAIO,WORKIO
```

Parameter Statement Processing

This section discusses the processing order of Adabas Review parameter statements and the effect of that order on the contents of detail and summary reports.

Defining Reports

The most critical parameter statement is the REPORT statement. It indicates that all of the following statements up to the next REPORT statement define the contents of the current report.

Example:

```
REPORT   TYPE=DETAIL
INCLUDE  ...
DISPLAY  ...
```

```
REPORT   TYPE=SUMMARY
EXCLUDE  ...
DISPLAY  ...
AVERAGE  ...
SUM      ...
```

```
REPORT   TYPE=DETAIL
DISPLAY  ...
```

There are three distinct reports: the first and third reports are detail reports; the second report is a summary report. Each report includes/excludes different input records; different statistics are compiled and printed for each report.

Multiple Parameter Statements

A parameter statement containing multiple parameter entries can be entered as one continuous statement or as several statements, each containing one or more parameter entries. For all parameter statements except the INCLUDE, EXCLUDE, and DISPLAY statements, either method will produce the same results.

Examples:

1. Print the sum for the values in the data fields COMMANDS, ASSOIO, and DATAIO on the report.

   ```
   SUM    COMMANDS, ASSOIO, DATAIO
   ```

2. Example 2 produces the same results as example 1:

```
SUM    COMMANDS
SUM    ASSOIO
SUM    DATAIO
```

INCLUDE/EXCLUDE Statements

INCLUDE/EXCLUDE parameter statements with multiple parameters have different logical meanings based on whether the parameters are entered as separate statements or as a single continuous statement.

Examples:

1. In this example, logical AND is implied between the SEQUENCE parameter and JOBNAME and RSP parameters. Input detail records must satisfy *all* three conditions in order to be selected for processing:

```
INCLUDE
SEQUENCE>100,JOBNAME=AAAAAAAA,RSP<150
```

2. In this example, logical OR is implied between each INCLUDE statement. Input detail records meet the selection criteria by satisfying *any* of the three conditions specified in the statements:

```
INCLUDE  SEQUENCE>100
INCLUDE  JOBNAME=AAAAAAAA
INCLUDE  RSP<150
```

3. In this example, logical AND is implied between the SEQUENCE parameter and the JOBNAME and DATE parameters entered on the first INCLUDE statement. Input detail records must satisfy *all* three conditions in order to be selected for processing. No further checking is done on the records satisfying these conditions; they are included in the processing regardless of whether they satisfy the conditions set by the remaining parameter statements:

```
INCLUDE
SEQUENCE>100,JOBNAME=AAAAAAAA,DATE<90201
EXCLUDE  RSP>0
INCLUDE  MONTH=12,DAY>15
EXCLUDE  FNR<101
```

Note:
It is not possible to use a field (parameter) more than once in a single INCLUDE statement; that is, in a logical AND operation.

Input detail records that do not satisfy the conditions specified on the first INCLUDE statement are checked against the selection criteria specified on the next EXCLUDE statement. All records satisfying this condition (i.e., RSP>0) are excluded from processing and no further checking is done on these records.

The detail records that have not been specifically included or excluded from processing by the first two parameter statements are checked against the selection criteria specified on the next INCLUDE statement. Again, logical AND is implied between the MONTH and DAY parameters entered on this statement, and both conditions must be met in order for records to be selected for processing. No further checking is performed on the records satisfying these conditions; they are included in the processing regardless of whether they satisfy the conditions set by the final EXCLUDE statement.

The remaining input detail records are then checked against the final EXCLUDE statement. All records satisfying the specified condition (i.e., FNR<101) are excluded from processing.

Note that all records not excluded by the final EXCLUDE statement are included in the processing. If the final statement is an INCLUDE statement, all records satisfying the specified conditions for inclusion are included in the processing.