Adabas Review

Using Batch Facilities

Version 4.3.2

September 2009

Adabas Review

This document applies to Adabas Review Version 4.3.2 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Table of Contents

1 Using Batch Facilities

Adabas Review may be used in batch processing mode to monitor resource usage for applications and databases by extracting data fields from the command log files generated by Adabas or Adabas Review.

To produce batch reports, the Adabas Review processor requires parameter statements and job control to run against the command log records. Report parameter statements can be manually created or they can be generated using the `GENCARD` command and subsequently edited, if required. Once the report statements are set up, the user submits the job stream for batch execution.

This part of the documentation describes how to use the `GENCARD` command to generate batch parameter statements. It also describes the statements required and presents guidelines for creating batch parameter statements without using the `GENCARD` command.

The Adabas Review Using Batch Facilities documentation is organized in the following topics:

2 Generating Batch Report Parameters

Note: This command is not available to users of Adabas Review under z/VM.

The GENCARD command creates batch report parameter statements from field and report processing information you provide through the Edit Report (ER) function. The GENCARD command also uses the target database information it obtains from the List Target Definitions (LT) function.

By using the GENCARD command to generate batch report parameters, you can run any online report in batch mode.

To use the GENCARD command

1 Access the Edit Report (ER) function.

2 Create the report definition in the same way that you create an online report definition; then save it by pressing PF5.

For more information, refer to section *Maintaining Report Definitions*.

3 Enter the command GENCARD or GC on the command line.

The Generate Report Definition Cards window appears as shown in the example below:

```
12:05:03                A D A B A S  -  R E V I E W                2003-07-07
                                Edit Report                         HUB=00221

        Report Name: COMMANDS BY HOUR_________________ DBID Monitored:222


+-------------------------------------------------------------------+
|   Field      Order  Sum   Min   Max   Avg   Pct   Rate   Round  |
+-------------------------------------------------------------------+
|   Generate Report Definition Cards                                 |
|   Please enter the following information for generation            |
|                                                                    |
|   File/DDNAME for output .... RVUCARD                              |
|   Report Definition Prefix .. ____________________________         |
|                                                                   |
+-------------------------------------------------------------------+
|  ________     __     _     _     _     _     _     _    _______ |
|  ________     __     _     _     _     _     _     _    _______ |
|  ________     __     _     _     _     _     _     _    _______ |
+--------------------------- Page 1-------------------------------+

 Command: gencard_____________________________________________________________
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Optns Exit        Save  Start  -     +          Rules Flds  Menu
```

4 On the line labeled "File/DDNAME for output", type the file or DD name for the sequential file where the batch report parameter statements are to be written.

The default name RVUCARD appears on this line.

5 On the line labeled "Report Definition Prefix", type the report name and press ENTER.

To generate batch parameters for a series of reports, you may use an asterisk for "wildcard" prefixing and suffixing.

For example, to generate batch parameters for all reports with names beginning with "T", enter T* on the Report Definition Prefix line. For all reports with names ending with "test", enter *test.

You receive a message that the generation process has been started; however, you do not receive a message when the process is complete.

6 To ensure that the batch parameters were generated, examine the sequential dataset where the parameters were to be written.

This chapter covers the following topics:

Sample Statements

The following is an example of the batch parameter statements generated by the GENCARD command for a sample report:

```
INPUT   FILETYPE=SEQUENTIAL,
        BUFFER-SEGMENTS=100,
        FILES=1,
        REVIEW-COMMANDS=NO
REPORT  TYPE=SUMMARY,
        TITLE='GENERATE EXAMPLE',
        PROGRAM=RD-00055,
        RESTART=Y,
        HISTORY-INTERVAL=15,
        MAXSTORE=8
DISPLAY SEQUENCE,CMD,RSP,TPUSERID,NATAPPL,NATPROG
SUM     IOS,CMDRESP,ADADURA
AVERAGE IOS,CMDRESP,ADADURA
PERCENT IOS,CMDRESP,ADADURA
INCLUDE RSP-(0,3,17,148),JOBNAME=(JOB1*-JOB3*)
COPY    FILETYPE=ADABAS,HUB=221,FNR=002,SVC=237
LOG     DSN=RVLOG,
        EXIT=LOGEX1,
        LOGIO=YES
```

Statement	Generated using information. . .
INPUT	about the target database from the List Target function (LT).
REPORT	from several sources: the report TYPE, TITLE, and PROGRAM from the List Report (LR) function; the other parameters from the Report Options screen of the Edit Report (ER) function.
DISPLAY	about the control break fields for summary reports or the data fields to be printed for detail reports from the Edit Report and Report Options screens of the Edit Report (ER) function.
SUM AVERAGE PERCENT	about data fields from the Edit Report screen of the Edit Report (ER) function.
INCLUDE	from the Report Processing Rules screen of the Edit Report (ER) function.
COPY	from the history parameters of the Report Options screen.
LOG	from the logging parameters of the Report Options screen.

For a more detailed description of these statements, refer to the section ***Parameter Statements***.

3 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

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 dataset 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.

Statement	Use
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

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

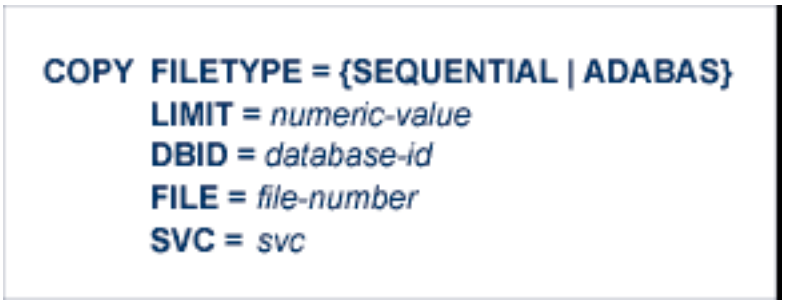
The AVERAGE statement is applicable to summary reports only. Average values are calculated only for the valid data fields specified in this statement. Average values are summarized and printed on the summary report at each control break defined by the parameter statement for all data fields for which 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



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 dataset or an Adabas file.

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

- **Copy detail records to a sequential dataset.**
  The user can select subsets of detail log records and copy them to different sequential datasets. These datasets 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 dataset 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

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

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.

"n.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.

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
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	◎	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 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.

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.

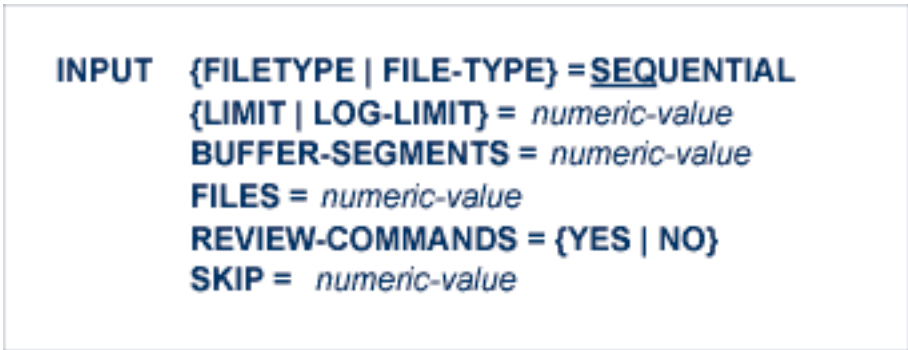
Note: The same field name may not be used more than once for a particular INCLUDE statement.

```
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

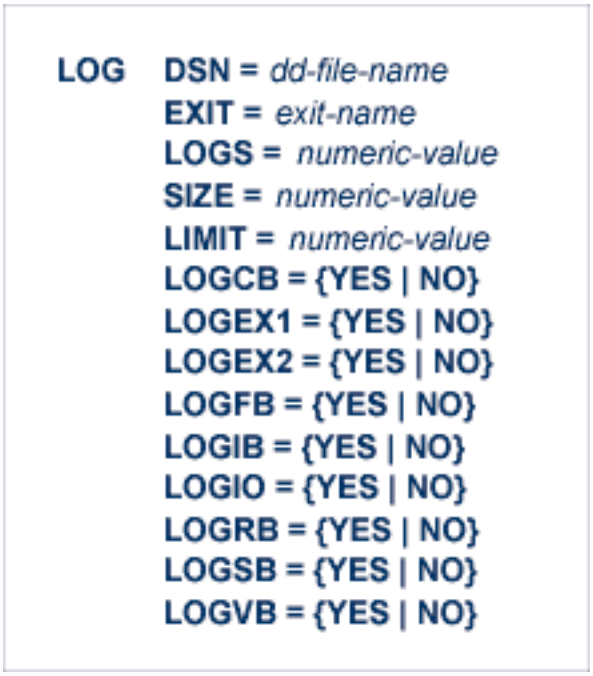


—where FILES is used only under VSE/ESA.

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 VSE/ESA, 400 (or 50K) for z/VM or BS2000, or 2000 (or 250K) for OS/390 or z/OS.
FILES	(VSE/ESA 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.

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 dataset is closed and before the next command log dataset is opened.
LOGS	the number of command log datasets for the report.
SIZE	the maximum number of blocks allocated to the command log dataset. When the SIZE parameter value is reached, the exit specified in the EXIT parameter is called, and the next command log dataset 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.

Parameter	Specifies . . .
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 = numeric-value
        BREAK = {YES | NO}
        DISPLAYBY = {SORTED | SUMFIELD | USAGE | SORTEDDE | DATETIME}
        ENTRIES = numeric-value
        HISTORY-INTERVAL = numeric-value
        LIMIT = numeric-value
        {LINESIZE | WIDTH} = numeric-value
        MAXSTORE = numeric-value
        {PAGESIZE | PAGE-SIZE} = numeric-value
        PRINT = {YES | NO}
        PROGRAM = name-of-display-program
        REPORT-EXIT = detail-report-user-exit-name
        RESTART = {YES | NO}
        {SKIP | SKIPPING} = {1 | 2}
        {SPACE | SPACING} = numeric-value
        SUMMARY-EXIT = summary-report-user-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 . . .
TYPE	whether the report to be generated is a detail or summary report. This parameter must be specified; there is no default.
ADALIMIT	a minimum command count (number of times the command was processed) for printing. For example, if ADALIMIT=100, only entries with a command count of 100 or higher are printed.
BREAK	whether subtotals are printed at control breaks or suppressed.
DISPLAYBY	the order in which the data is displayed: **SORTED** ascending order by control break. **SUMFIELD** descending order by the first field marked as a sum field. **USAGE** descending order by usage (e.g., most used to least used command). **SORTEDDE** descending order by control break. **DATETIME** ascending order by the start date and time of the control break interval.

Keyword	Specifies . . .
	LINEAR (not available for batch reporting)
ENTRIES	the maximum number of entries that a report will process, thus restricting the amount of data collected by the summary report.
HISTORY-INTERVAL	the frequency (in minutes) with which history data is written.
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.
{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. The default value is 132 characters.
MAXSTORE	the maximum amount of storage (in kilobytes) available for the report.
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.
PRINT	whether the report will be printed at database termination.
PROGRAM	the name of the display program to be used if the report results are displayed online.
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.
RESTART	whether the report is reactivated after the history interval is reached, the report is closed, or the report MAXSTORE limit is exceeded.
SKIP	whether to single-space or double-space the detail lines on the output report. The default is to single-space the detail lines.
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.
SUMMARY-EXIT	he 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.
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.
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.
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.
WRAP	whether wrapping (i.e., reusing data elements for the report) will occur. By default, wrapping will not occur.

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.

4 Batch Processor Job Control Requirements

This chapter covers the following topics:

For OS/390 or z/OS Environments

The job shown below is contained in member REVIEWB of the Adabas Review source library and can be used to execute the Adabas Review processor in batch.

To modify the job before submitting it

1 Change "vrs" to the current version, revision, and system maintenance level of the product.

2 Modify the job card, if necessary.

3 Modify the DD statement for RVUSEQ as necessary. Point to a command log file generated by Adabas or Adabas Review.

4 Add any additional RVUPRTxx or RVUCOPxx DD statements as necessary depending on your report definitions (where "xx" is a value 01 through 99).

```
//REVIEWB  JOB
(LOCATION),`REVIEW',MSGCLASS=X,CLASS=A
//*
//REVIEWB  EXEC PGM=REVIEWB,REGION=512K
//STEPLIB  DD DISP=SHR,DSN=REVIEW.Vvrs.LOAD
//         DD DSN=SHR,DSN=ADABAS.Vv.LOAD
//*
//RVUSEQ   DD DSN=SHR,DSN=ADABAS.COMMAND.LOG.FILE.
//         DCB=(RECFM=VB,BLKSIZE=10000)
//RVUFLD   DD DSN=SHR,DSN=REVIEW.Vvrs.SOURCE(USRPARM)
//*
//RVUPRT00 DD SYSOUT=X,LRECL=80
//RVUPRT01 DD SYSOUT=X,LRECL=133
//RVUPRT02 DD SYSOUT=X,LRECL=133
//SYSUDUMP DD SYSOUT=X
//TRACEOUT DD SYSOUT=X,LRECL=133
//*
//RVUPARM  DD *
INPUT   FILETYPE=SEQUENTIAL,LIMIT=1000
*
REPORT  TYPE=SUMMARY,TITLE=`SAMPLE REPORT'
AVERAGE DURATION,ASSO-IO,DATA-IO,CMDRESP
MINIMUM DURATION,ASSO-IO,DATA-IO,CMDRESP
MAXIMUM DURATION,ASSO-IO,DATA-IO,CMDRESP
DISPLAY JOB
*
/*
```

The following DD statements are required, or optional where noted, for executing the Adabas Review processor in interactive or batch mode:

DD Statement	Description
RVUPARM	A dataset of control statements that specify input report parameters. These statements can be generated by the GENCARD command and copied from the resulting RVUCARD output into the RVUPARM dataset. LRECL=80.
RVUSEQ	Sequential dataset containing command log records: RECFM=VB,LRECL=9996,BLKSIZE=10000
RVUCOPxx	(Optional) Copied output logs; same format as RVUSEQ (where "xx" is 01-99).
RVUPRTxx	Review logical printers (where "xx" is 01-99). LRECL is required. It may be in the range of (72-4080); LRECL=133 is typical.
RVUALT	Alternate sequential file used to save history information if the Adabas Review processor, either interactively or in batch, receives an Adabas response code 148 (Adabas not active) when attempting to save history data. This file should be allocated using the following DCB attributes: RECFM=VB,LRECL=9996,BLKSIZE=10000
RVUAUT1	(Optional in batch mode) Parameter card images for autostarted reports; LRECL=80.
RVUAUT2	(Optional in batch mode) Parameter card images for autostarted reports; LRECL=80. **Note:** Adabas Review uses two parameter files for the report definition control statements and alternates between them by writing to the older file. See the section Autostarted Reports on page .
RVUFLD	Parameter card images; LRECL=80. Parameters to describe user-defined fields.

For VSE/ESA Environments

The job shown below is contained in member REVIEWB of the Adabas Review source library and can be used to execute the Adabas Review processor in batch.

To modify the job before submitting it

1 Change "vrs" to the current version, revision, and system maintenance level of the product.

2 Modify the job cards, if necessary.

3 Modify the job control statement for RVUSEQ as necessary. Point to one of the command log files.

4 Add any additional RVUPRTxx or RVUCOPxx job control statements as necessary, depending on your report definitions (where "xx" is a value 01 through 99).

```
//  JOB REVIEWB                              sample
Review job
//  EXEC  PROC=REVvrs                        Review private libraries
//  ASSGN SYS005,SYSIPT                      RVUPARM - cards
//  ASSGN SYS006,DISK,VOL=VVVVVV,SHR         RVUSEQ - tape
//  ASSGN SYS007,IGN                         MAY be IGN for batch
//  ASSGN SYS020,PRINTER                     RVUPRT0 - printer
//  ASSGN SYS021,CUU                         RVUPRT1 - printer
//  DLBL RVUSEQ, 'ADABAS.Vvr.COMMAND.LOG'    RVUSEQ - command
log
//  EXTENT SYS006,VVVVVV
//  EXEC REVIEWB,SIZE=(AUTO,64K)
INPUT    FILETYPE=SEQUENTIAL,LIMIT=1000
REPORT   TYPE=SUMMARY,TITLE='SAMPLE REPORT'
AVERAGE  DURATION
MINIMUM  DURATION,ASSO-IO,DATA-IO,CMDRESP
MAXIMUM  DURATION,ASSO-IO,DATA-IO,CMDRESP
DISPLAY  JOB
/*
```

Note: The logical units shown in the example above may be reassigned if there are conflicts with your site-specific logical units. Refer to the Adabas Review Installation and Operations (VSE/ESA) documentation for more information.

The following job control statements are required, or optional where noted, for executing the Adabas Review processor in interactive mode or batch mode:

Job Control Statement	Logical Unit(s)	Description
RVUPARM	SYS005	Parameter card images; 80-byte records. RVUPARM is a dataset of control statements that specify input report parameters. These statements can be generated by the `GENCARD` command and copied from the resulting RVUCARD output into the RVUPARM dataset.
RVUSEQ	SYS006	Sequential dataset containing command log records: Record Format = VB, Record Length = 9996, Block Size = 10000.
RVUCOPx	SYS031-39	(Optional) Copies output logs; same format as RVUSEQ (where "x" is 1-9).
RVUPRTx	SYS020-29	Review logical printers (where "x" is 0-9).
RVUALT	SYSxxx*	Alternate sequential file used to save history information if the Adabas Review processor, either interactively or in batch, receives an Adabas response code 148 (Adabas not active) when attempting to save history data. This file should be allocated using the job DBFILES in the Adabas Review source library. * May be any unused value.

Job Control Statement	Logical Unit(s)	Description
RVUAUT1 / RVUAUT2	SYS007	(Optional in batch mode) Parameter card images for autostarted reports. Adabas Review uses two parameter files for the report definition control statements and alternates between them by writing to the older file. **Note:** SYS007 may be assigned to IGN when running Adabas Review in batch mode.

For BS2000 Environments

The job shown below is contained in member P.REVBATCH of the Adabas Review source library and can be used to execute the Adabas Review processor in batch.

To modify the job before submitting it

1 Set &ADAL to the Adabas Library

2 &BATCH to the output file prefix

3 &CLOG to the command log generated by Adabas or Adabas Review &REVL to the Adabas Review Library

4 modify the parameters following the '/STA-EDT' card to those required for the job

```
/BEGIN-PROC C,PROC-PAR=( -
/ &ADAL=$SAG.ADABAS.MOD,-
/ &BATCH=BATCH,-
/ &CLOG=$SAG.DB00099.CLOGR1,-
/ &DUMP=YES,-
/ &REVL=$SAG.REVIEW.MOD -
/ ),ESC-CHAR='&'
/REMARK
/REMARK          ***************************************************
/REMARK          *    START REVIEW BATCH                           *
/REMARK          ***************************************************
/REMARK
/MOD-TEST        DUMP=&DUMP
/DEL-FI #RVUPARM
/SET-JOB-STEP
/SET-FILE-LINK    EDTSAM,#RVUPARM ,REC-FORM=F(REC-SIZE=80)
/MOD-J-SW ON=(4,5)
/ASS-SYSDTA *SYSCMD
/STA-EDT
INPUT  FILETYPE=SEQUENTIAL,LIMIT=1000
REPORT TYPE=SUMMARY,TITLE='SAMPLE REPORT'
AVERAGE DURATION,ASSO-IO,DATA-IO,CMDRESP
MINIMUM DURATION,ASSO-IO,DATA-IO,CMDRESP
MAXIMUM DURATION,ASSO-IO,DATA-IO,CMDRESP
```

```
DISPLAY JOB
@W '#RVUPARM' O
@H
/SET-JOB-STEP
/MOD-J-SW OFF=(4,5)
/SET-JOB-STEP
/ASS-SYSLST   L.REV&BATCH..L
/ASS-SYSDTA *SYSCMD
/SE-F-LI         RVUAUT1,*DUMMY
/SE-F-LI         RVUAUT2,*DUMMY
/SE-F-LI         RVUPRT00,&BATCH..RVUPRT00
/SE-F-LI         RVUPRT01,&BATCH..RVUPRT01
/SE-F-LI         RVUPRT02,&BATCH..RVUPRT02
/SE-F-LI         RVUSEQ,&CLOG
/SE-F-LI         RVUPARM,#RVUPARM
/SE-F-LI         COMPRINT,#COMPRINT,REC-FORM=F(REC-SIZE=80)
/REMA
/SE-F-LI         DDLIB,&ADAL
/SE-F-LI         BLSLIB00,&REVL
/SET-JOB-STEP
/STA-PROGRAM   (&REVL,REVBATCH),RUN-M=ADV(A-L=YES)
/ASS-SYSLST *PRIM
/ASS-SYSDTA *PRIM
/END-PROC
```

The following Link statements are required (or are optional where noted), for executing the Adabas Review processor in interactive or batch mode:

Link Statement	Description
RVUPARM	Parameter card images; REC-FORM=FIXED(REC-SIZE=80). RVUPARM is a dataset of control statements that specify input report parameters. These statements can be generated by the GENCARD command and copied from the resulting RVUCARD output into the RVUPARM dataset.
RVUSEQ	Sequential dataset containing command log records
RVUCOPxx	(Optional) Copied output logs; same format as RVUSEQ (where 'xx' is 01-99).
RVUPRTxx	Review logical printers (where 'xx' is 01-99). Sequential file with REC-FORM=FIXED(REC-SIZE=133).
RVUALT	Alternate sequential file used to save history information if the Adabas Review processor, either interactively or in batch, receives an Adabas response code 148 (Adabas not active) when attempting to save history data.
RVUAUT1	(Optional in batch mode) Parameter card images for autostarted reports; REC-FORM=FIXED(REC-SIZE=80).
RVUAUT2	(Optional in batch mode) Parameter card images for autostarted reports; REC-FORM=FIXED(REC-SIZE=80).

Note: Adabas Review uses two parameter files for the report definition control statements and alternates between them by writing to the older file. See the section Autostarted Reports. RVUFLD Parameter card images; REC-FORM=FIXED(REC-SIZE=80). Parameters to describe user-defined fields.

For z/VM Environments

The Adabas Review processor is executed in batch mode using the REVBATCH EXEC provided on the Adabas Review installation tape. The only parameter required is the name of the file containing the Review input parameters. The FILETYPE of the input parameters must be "REVPARMS". For example, the following runs Adabas Review in batch mode using the report parameters found in "SAMPLE1 REVPARMS":

```
REVBATCH SAMPLE1
```

The following filedef statements are required, or optional where noted, for executing the Adabas Review processor in interactive mode or batch mode:

Filedef Statement	Description
RVUPARM	Parameter card images; LRECL=80.
RVUSEQ	Sequential dataset containing command log records: (RECFM VB LRECL 9996 BLOCK 10000)
RVUCOPxx	(Optional) Copied output logs; same format as RVUSEQ (where "xx" is 01-99).
RVUPRTxx	Review logical printers (where "xx" is 01-99). LRECL is required. It may be a value in the range 72-408; LRECL=133 is typical.
RVUALT	Alternate sequential file used to save history information if the Adabas Review processor, either interactively or in batch; receives an Adabas response code 148 (Adabas not active) when attempting to save history data. This file should be allocated using the following DCB attributes: (RECFM VB LRECL 9996 BLKSIZE 10000)
RVUAUT1 RVUAUT2	(Optional in batch mode) Parameter card images for autostarted reports; LRECL=80. **Note:** Adabas Review uses two parameter files for the report definition control statements and alternates between them by writing to the older file. See the section Autostarted Reports on page .

5 Using Adabas Review in Batch Natural

The functions described earlier to start and view Adabas Review reports are used mainly in an online environment. You may also perform these functions in a batch job to obtain statistics on a batch Adabas job as output from the job itself.

Note: If you are trying to start a report in hub mode using batch Natural , you must issued the MENU HUB=*hubid* command prior to issuing the START command for the report.

This chapter covers the following topics:

Example

A batch Natural job (PAYROLL1) is run each night. When the job is completed, Adabas summary statistics are requested to determine the number of Adabas calls issued by the job, the files accessed, the type of Adabas commands issued, and a summary of ASSO, DATA, and WORK IOs for each command type within each file.

To implement the request for Adabas summary statistics

1 Ensure that the Review load library is in the STEPLIB concatenation.

2 Define the following report definition and related processing rules to Adabas Review:

```
20:47:45                A D A B A S  -  R E V I E W              2003-07-07
                              Edit Report                        LOCL=00009


 Report Name: TRACE PAYROLL REPORT_____________  DBID to Monitor:____9


 +-------------------------------------------------------------------+
 |  Field      Order   Sum   Min   Max   Avg   Pct   Rate   Round  |
 |-------------------------------------------------------------------|
 | FILE____     _10     _     _     _     _     _     _    _______  |
 | CMD_____     _20     _     _     _     _     _     _    _______  |
 | COMMANDS     ___     X     _     _     _     _     _    _______  |
 | ASSOIO__     ___     X     _     _     _     _     _    _______  |
 | DATAIO__     ___     X     _     _     _     _     _    _______  |
 | WORKIO__     ___     X     _     _     _     _     _    _______  |
 | ________     ___     _     _     _     _     _     _    _______  |
 | ________     ___     _     _     _     _     _     _    _______  |
 | ________     ___     _     _     _     _     _     _    _______  |
 +---------------------------- Page 1-------------------------------+

 Command: _____________________________________________________________
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Optns Exit        Save  Start  -     +          Rules Flds  Menu
```

```
20:50:02                A D A B A S  -  R E V I E W              2003-07-07
                           Report Processing Rules               LOCL=00009

                             TRACE PAYROLL REPORT

 +-------------------------------------------------------------------------+
 |  Field    Op        Value                                       And/Or  |
 |-------------------------------------------------------------------------|
 | JOBNAME_  =  PAYROLL1_____________________________________________  ___  |
 | ________  __ ____________________________________________________  ___  |
 | ________  __ ____________________________________________________  ___  |
 | ________  __ ____________________________________________________  ___  |
 | ________  __ ____________________________________________________  ___  |
 | ________  __ ____________________________________________________  ___  |
 | ________  __ ____________________________________________________  ___  |
 | ________  __ ____________________________________________________  ___  |
 | ________  __ ____________________________________________________  ___  |
 | ________  __ ____________________________________________________  ___  |
 +---------------------------------- Page 1-----------------------------------+

 Command: _____________________________________________________________
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Optns Exit        Save  Start  -     +               Flds  Menu
```

3 Save the report definition.

4 Change the PAYROLL1 job stream if necessary to produce the required results shown below:

```
LOGON SYSREVDB                      <---logon to
Review Natural library
START TRACE.PAYROLL.REPORT dbid     <---start Review report
LOGON PAYROLL
*
* PAYROLL1 NATURAL STREAM
*
LOGON SYSREVDB                      <---logon to Review Natural
library
VIEW TRACE.PAYROLL.REPORT dbid      <---to display report results
DELREP TRACE.PAYROLL.REPORT dbid    <---to delete report results
FIN
```

Note: If you do not specify a DBID, Adabas Review uses the DBID corresponding to LFILE 241.

Processing History Data in Batch Natural

You may also use batch Natural programs to display, purge, or compress history data from the Adabas Review repository. Adabas Review provides sample jobs to perform these tasks.

To use these jobs

1 Point the LFILE to the DBID and FNR corresponding to the Adabas Review repository containing the history data.

2 Add the report name and date range.

3 Enter any embedded blanks in the report name as periods.

To display history data

- Use the HISTVIEW job.

  The following Natural statements are contained in the display history data job stream:

```
LOGON SYSREVDB                      <--logon to
Review Natural library.
HISTORY A.HISTORY.REPORT start-date end-date  <--report name and
dates.
FIN
/*
```

To delete history data

- Use the HISTDEL job.

  The following Natural statements are contained in the delete history data job stream:

```
LOGON SYSREVDB                      <--logon to
Review Natural library.
DELHIST A.HISTORY.REPORT start-date end-date  <--report name and
dates.
FIN
/*
```

To compress history reports

- Use the HISTCOMP job.

  The following Natural statements are contained in the compress history job stream:

```
LOGON SYSREVDB                        <--logon to
Review Natural library.
COMPRESS A.HISTORY.REPORT start-date end-date et-factor <--report
info.
FIN
/*
```

Unloading and Restoring Report Definitions

Review report definitions can be unloaded to a sequential dataset for backup, archive or for the purpose of moving the definition to a different Review system. There are two batch Natural programs to accomplish this, ULDREP2 and LODREP2. ULDREP2 unloads a single report definition to a sequential dataset as defined by the Natural work file CWFWK01. LODREP2 restores the definition from the same sequential dataset.

The syntax for ULDREP2 is:

```
ULDREP report.name
```

where REPORT.NAME is the report name with periods between each word in the report name.

The syntax for LODREP2 is:

```
LODREP report.name
```

The job control statements to run ULDREP2 and LODREP2 are the same as other Review batch Natural jobs such as START, VIEW and DELHIST.

Index