Advantage™ VISION:Results™ for z/OS

Getting Started

r6

Contents

Chapter 1: Welcome

Chapter 2: Features

Chapter 3: Commands and Statements

Chapter 4: Structure of a VISION:Results Program

Chapter 5: Programming Examples

Chapter 6: Structured and USERDEFAULT Mode Programming

Appendix A: Definition of Terms

Appendix B: Numeric and Packed Key Considerations

Index

Chapter 1: Welcome

Advantage™ VISION:Results™ for z/OS is an information management and report generator tool with an easy-to-use, English-like programming language that makes it convenient to enter statements at terminals. VISION:Results offers these key features:

- Multiple file and database-type support
- Use of existing data definitions
- Extensive data selection, analysis, and handling capabilities

VISION:Results supports all IBM file structures and access methods, and is available for z/OS and VSE/ESA operating systems.

VISION:Results provides both programmers and non-programmers with the capability to program data processing tasks in a fraction of the time that it would take to employ traditional programming languages such as COBOL or PL/I.

Although VISION:Results has the power and flexibility of a full-function language, it is a supplementary tool and should not replace an installation's standard programming language.

Note: Advantage VISION:Results will be referred to as VISION:Results throughout this guide.

Simple Examples

The following examples illustrate the power of VISION:Results to create simple programs using just a few statements.

Following are a few examples of how to write a program to do a task:

To copy a file

```
FILE INFILE
FILE OUTFILE OUTPUT FROM INFILE FB 200 4000
```

To copy a file and drop a bad record

```
FILE INFILE

  ACCOUNT 5 1 NU
FILE OUTFILE OUTPUT FROM INFILE FB 200 4000
IF ACCOUNT EQ 52213 REJECT ENDIF
```

To produce a simple report of names and addresses

```
FILE INFILE
  NAME 30 6   ADDRESS 30
LIST NAME ADDRESS
```

To print a file in hexadecimal and graphics representation

```
OPTION HEXPRINT
FILE INFILE
```

To print the first 50 records of a file in graphics representation

```
FILE INFILE  COUNT INCOUNT
IF INCOUNT EQ 51 STOP ENDIF
PRINT INFILE
```

Even though these examples illustrate basic VISION:Results functions, VISION:Results is capable of handling many more demanding and complex tasks. For more examples of how to use VISION:Results, see Chapter 5: Programming Examples.

Environment

This section describes the packaging and environmental considerations for r6.

IBM Operating Systems Supported

Separate versions of the system exist to support the particular IBM operating system in use at an installation. The z/OS version supports all z/OS operating systems. The VSE version supports VSE/ESA. The CMS version supports VM/XA and VM/ESA.

Available Interfaces

You can use the following interfaces with VISION:Results:

- Advantage™ VISION:Results™ Interface™ to CA-IDMS/DB
- Advantage™ VISION:Results™ Interface™ to DB2
- Advantage™ VISION:Results™ Interface™ to IMS/DLI
- Advantage™ VISION:Results™ Interface™ to SQL/DS

Documentation

The following VISION:Results documentation set is available on the CA web site at http://supportconnect.ca.com.

Guide Name	PDF File Name	Description
Advantage VISION:Results for z/OS Release Summary	RSRLS060.pdf	Documents new features and changes to existing features for r6.
Advantage VISION:Results for z/OS Getting Started	RSGSM060.pdf	Contains fundamental descriptions of VISION:Results and how to use it.
Advantage VISION:Results for z/OS Installation Guide	RSINM060.pdf	Describes how to install VISION:Results and customize it with the DYLINSTL macro.
Advantage VISION:Results for z/OS Reference Summary	RSRFS060.pdf	A compendium of information about VISION:Results syntax, reserved words, programming tips, and definitions.
Advantage VISION:Results for z/OS Reference Guide	RSREF060.pdf	Contains information specific to the use and operation of VISION:Results. Commands, syntax, procedures, and report generation are discussed in detail. Includes chapters on VSAM, ISAM, and BDAM file processing, letter writing, as well as MATCH/MERGE and statistical features. **Note:** IBM withdrew support for ISAM beginning with z/OS r1.7. Therefore, VISION:Results cannot support access to ISAM data sets under operating systems after z/OS r1.6.

Guide Name	PDF File Name	Description
Advantage VISION:Results for z/OS Toolkit Reference Guide	RSTLK060.pdf	Contains information about the Macro facility, FREEZE/RESTORE, comma separated values (CSVRSLT), DYLPDS, DYLTABLE, PCFILE/PCWRITE, and APPC/MVS. Also contains information about executing Advantage™ VISION:Report® from VISION:Results, label generation, and library functions.
Advantage VISION:Results for z/OS Messages and Codes	RSMSG060.pdf	Contains validation, error handling, and DUMP information. Contains codes, messages, and explanations. Error analysis includes the condensed error analysis listing, completion codes, using cross-references, and so on.

Contacting CA Technical Support

For online technical assistance and a complete list of locations, primary service hours, and telephone numbers, contact Technical Support at http://ca.com/support.

Chapter 2: Features

This chapter describes the features of VISION:Results.

I/O, Data Selection, and Manipulation

- Sequential, VSAM (KSDS, ESDS, and RRDS), BDAM, source statement library (VSE), or partitioned data set (z/OS) file support.
- Fixed, variable, variable-spanned (z/OS), or undefined format file support.
- Instream data, tape, or disk (2311, 2314, 3330, 3340, 3350, 3375, 3380, 3390, 3540) and FBA (Fixed Block Architecture) support.
- Unlimited sequential file processing. Eight files in or eight files out with the automatic cycle. Unlimited files if using I/O by the EXIT option.
- File print (for example, hexadecimal, character, hexadecimal and character).
- Full sorting capabilities.
- Data translation—blanks to zeros, ASCII to EBCDIC, and so on.
- Character, zoned decimal, packed, or binary data handling.
- Data conversion—pack, unpack, binary, or CSV conversion.
- Bit testing—test for bits “on”.
- Bit manipulation—move with offset, move zone, and so on.
- Add, subtract, multiply, or divide data in zoned decimal, packed, or binary format. Automatic decimal alignment.
- Exponentiation—raise a value to a power. positive, negative, integer, and decimal exponents are supported.
- Record selection—include or exclude records based on a variety of criteria.
- Random and interval selection—perform nth interval data selection or random selection.
- Record reformatting—rearrange, omit, or convert to create any type of record that you want.
- Data editing.

- Table and array processing—table and array handling are built into VISION:Results. The size is limited only by available memory. You can pass the table or array to subsequent requests in a multiple-request program without an intermediate file.
- Indexing feature—allow address modification for table loading and retrieval or variable record accessing and creating.
- User exit routines—allow access to databases, special data manipulation, and so on.
- Automatic printing of I/O statistics at end of run.

Report Generation

- Automatic composition—allow VISION:Results to handle the report layout, including determining column heading locations and print field locations automatically.
- Left and right alignment—you can override automatic centering of individual column headings to allow left or right alignment of the column heading.
- One to nine heading or footing lines.
- One to nine column heading lines.
- Automatic titling and page ejection.
- Multi-line reporting—you can print 1-99 lines for each record.
- User-controlled page width (default is 132; up to 204 for z/OS and VSE).
- User-controlled page length (default is 55 lines per page).
- User-controlled line spacing (default is single spacing).
- Automatic totaling (regular or cumulative).
- Control break totaling.
- Automatic control break determination—up to six levels plus final totals.
- Editing of output fields, including decimal alignment, rounding of fields (if specified), date editing, and editing of social security numbers.
- Output reports in HTML format.
- Optional print suppression of any field where the value has not changed.

Letter Writing

- Generate an unlimited number of personalized letters and memos, incorporating variable data that you can edit.
- Create unique letters by specifying ranges of detail lines. You can use preprinted forms on both impact and laser printers.

Data Processing Functions

- Copy files—you can create files of any supported organization or format from a file of the same or different format. I/O counts print automatically at the end of a run.
- Macro facility—you can use, create, and modify macros to meet user specifications.
- Print files—you can print files of any supported organization in graphics, hexadecimal and graphics, or hexadecimal only format. Column indicators are printed at the top and the bottom of the page to enable quick location of data fields. You can print every record or selected records.
- Convert files—you can write files from one device to another. VISION:Results can aid in conversions from one disk device to another (3350 to 3380) or from one system to another (VSE to z/OS).
- Production or test file generation—VISION:Results can create files needed for systems being tested or implemented using instream data or existing files as input.
- Update files—a master file of any supported organization can be updated by one or more transaction files, or no file updating can be done under the user's control.
- Sort files—you can sort and process files of any organization, instream, tape, or disk.
- Update-in-place VSAM files—you can update records on an existing DAM, BDAM, or VSAM file or you can add new records.
- Validate input—you can process input data and fields checked for validity. You can make tests for positive, numeric, certain values or value ranges, and so on.
- Match and merge files—you can match or merge files of the same or different organization on the same or different devices.
- Compose and print reports—you can use the automatic composition feature of VISION:Results to automatically format a report.
- Print preprinted forms—you can use the report writer, with its multi-line capability, to print on invoices, paychecks, and other preprinted forms. You can specify fixed print locations.
- XML support—using XML, you can export VISION:Results XML formatted data from the mainframe to your PC. XML allows you to describe, define, create, and transmit data in any format.

Linear Regression, Trend Line Analysis, and Scatter Diagram

- These functions calculate data, which allows you to draws statistical conclusions regarding the relationship between dependent and independent variables.
- VISION:Results plots actual (x,y) values on a scatter diagram, and defines the best-fitting line through these points.

Extensive Error Analysis and Correction

- During execution, if a data error occurs caused by blank data that should be zeros or data being divided by zero, VISION:Results corrects these abnormal conditions. If VISION:Results cannot correct an error, information about the error and the corresponding data area is printed.

Chapter 3: Commands and Statements

This chapter provides an overview of the more frequently used commands and statements in a VISION:Results program. It is not intended as a complete list or description, but should serve to present the basic elements.

Some VISION:Results commands stand alone in the program; they do not need keywords to modify or define them. Examples of these are NEWPAGE, FIN, and STATEON. Other commands, however, must be used in a statement. Statements consist of a command followed by a keyword that modifies the command. For example, the FILE command is always used in statement form.

For definitions of commands not described in this section, as well as information about keywords, edit codes, and data types, see Appendix A: Definition of Terms.

Program Control Commands

This section describes the program control commands.

OPTION Command

The OPTION command is used primarily to override VISION:Results defaults and supply information about the installation's operating environment.

You can code information in the OPTION statement to:

- Verify syntax (OPTION VERIFY).
- Alter free-form start and stop columns (OPTION COLUMNS).
- Request a data name cross-reference listing or a data map by area and location (OPTION XREFA, OPTION XREF, OPTION DMAP).
- Turn on the file print feature (OPTION PRINT, OPTION LCPRINT, OPTION HEXPRINT, OPTION HEX, and OPTION REPORTFILE (VSE only)).
- Supply variable data to the program (OPTION DATA 'literal').
- Override lines per page on program or file print (OPTION nn LONG).

- Override VSE work file device type (3330), SYS number, and file name defaults (OPTION DISK nnnn, OPTION FBA, OPTION SYSnnn, OPTION SYSnnn 'file name').
- Allow non-unique data names in a program (OPTION QUALIFIERS, OPTION QLF).
- Invoke the structured or USERDEFAULT programming modes (OPTION STRUCTURED, OPTION STRUCTURED2, OPTION USERDEFAULT).

If coded, OPTION must be the first statement in the program.

```
OPTION HEXPRINT FILEIN 60 LONG
```

For more information about using the OPTION command, see the *Advantage VISION:Results for z/OS Reference Guide*.

STATEON, STATEOFF Commands

You can code these commands anywhere throughout your program either to discontinue printing of program statements or to resume printing of program statements.

```
FILE FILEIN FB 80 800
   STATEOFF
     ACCNUM 5 NU
     ACCNAME 20
     .
     .
   STATEON
```

NEWPAGE Command

The NEWPAGE command forces page ejection to occur on the source statement listing.

FIN Command

The FIN command concludes the VISION:Results program statements. It is required when instream data follows the program.

Definition Commands

This section describes the definition commands.

FILE Command

A FILE statement is required for every user file that is being processed in the program. This statement provides information about file attributes and processing mode.

```
FILE ARFILE INPUT FB 352 3520
FILE ARFILEO OUTPUT FROM ARFILE FB 352 3520
```

TABLE Command

Defines the attributes of a 1-dimensional table. You can define up to 255 tables, using syntax similar to that required when defining a file.

```
TABLE DEPTTB F 33 ENTRY TBLENT STATUS TBSTAT
TABLE TAGTABLE F 10 STATUS TTBSTAT ENTRY TTBENT
  KEYLOC 5 KEYLEN 6
```

ARRAY Command

Defines the attributes of a 2- to 4-dimensional array. You can define up to 255 arrays, using syntax similar to that required when defining a file.

```
ARRAY EMPLIST F 29 DIM (50 3) STATUS TBLSTAT
ARRAY HISTORY F 30 DIM (10 12) STATUS HISTAT
```

SAMPLE Command

Defines the type of test data selection (interval or random). Use this function in the IF SAMPLING command.

```
SAMPLE 01 INTERVAL 10 5 100
SAMPLE 02 RANDOM 2%
```

LTH and LTD Commands

LTH defines the heading of a letter format in a VISION:Results program; LTD defines the body of the letter. You can append LTD with a numeric value from 1 to 99 to use the FROM-TO letter detail line option.

```
LTH 01 80 WIDE 50 LONG NOEJECT
LTD1
LTD2   (Body of the letter)
.
.
LTD3
ENDLTD
```

LINEAR, TREND, and SCATTER Commands

Commands that specify linear regression, trend line analysis, or scatter diagrams in the VISION:Results program. Use these functions in conjunction with the REGRESSION command.

```
LINEAR 01 XNAME SALES YNAME PROFIT
TREND 02 XNAME YEARS YNAME SALES
SCATTER 03 XNAME SALES YNAME PROFIT
  XSCATTER 2 FROM 0 BY .5 FOR 24
  YSCATTER 2 FROM 0 BY .2 FOR 18
```

Procedural Commands and Statements

Procedural commands or statements read or write data, select data, move and convert data, and perform arithmetic on data.

MOVE Command

Use the MOVE command to move data from one location in memory to another (for example, from an input record location to an output record location). If the send and receive fields have different attributes, data conversion and decimal alignment are handled automatically.

```
MOVE FIELDA TO FIELDB
MOVE 12345 TO FIELDC
```

Arithmetic Statements

Numeric, packed, or binary data with the same or different attributes can be added, subtracted, multiplied, divided, or raised to a power.

```
FIELDA=FIELDA+1
FIELDB=FIELDC*FIELDD/100
FIELDB=FIELDA**5**2
```

IF Command

Provides the ability to select data, or to process data based on certain conditions.

```
IF FIELDA EQ 'CA' GOTO PROCB ENDIF
IF FIELDB NE FIELDC REJECT ELSE ACCEPT ENDIF
IF RECCOUNT GT 100 STOP ENDIF

IF FIELDG NEGATIVE
   MOVE '*' TO COMMENT
   NEGCOUNT=NEGCOUNT+1
   GOTO LISTDET
ENDIF
```

GOTO Command

Transfers program control to a statement designated by a tag name.

```
GOTO PROCA

IF FIELDA GT FIELDB GOTO PROCB ELSE GOTO PROCC ENDIF
```

PERFORM Command

Executes a set of statements in another part of the program, then returns control to the statement following the PERFORM statement.

```
PERFORM PROCA TO PROCEXIT GOTO PROCC
```

CALL Command

Calls or invokes the external routine you specify. This allows you to do special processing on data or to read non-conventional files. You can write the external routine in Assembler, COBOL, FORTRAN, or PL/1. You can invoke as many separate routines as you require in any program.

```
CALL DATECONV USING JULDATE RETURNDATE
```

READ Command

Files can either be read automatically by VISION:Results, or you can issue READ commands in your procedure logic. A READ command is necessary if you are processing DAM, BDAM, or VSAM files in random, skip sequential, or sequential I/O mode.

```
READ FILEIN
```

WRITE Command

Files can either be written automatically by VISION:Results, or you can issue WRITE commands in your procedure logic. A WRITE command is necessary if you are processing DAM, BDAM, or VSAM files in random I/O or sequential I/O mode.

```
WRITE FILEOUT
```

LIST Command

Writes a line to the report that describes how the fields are to be edited, and in what order they are to be printed (from left to right). VISION:Results automatically composes the report based on what you code into this statement.

```
LIST ACCOUNT NAME BALANCE A
```

SORT Command

Sorts data before it is reported on or written. You can SORT sequential, DAM, BDAM, or VSAM files. Use this command to sort entire records, or just portions of records. You can sort at any point in your program.

```
SORT FILEIN USING DIV DEPT EMPNUM
SORT RECFLDS 50 USING FIELDA FIELDB
```

IF SAMPLING Command

Initiates interval or random selection using criteria supplied in the SAMPLE statement.

```
IF SAMPLING 01 ACCEPT ELSE REJECT ENDIF
```

LETTER Command

Prints letters defined in the LTH header and LTD text detail statements. You can also specify the FROM-TO line range option.

```
LETTER 01
LETTER 02 FROM 6 TO 10
```

REGRESSION Command

Initiates linear regression, trend line analysis, or scatter diagrams as defined in the LINEAR, TREND, or SCATTER statement. Optionally, it is followed by one of these keywords for documentation purposes.

```
REGRESSION TREND 02
REGRESSION LINEAR 01
```

MATCH Command

Matches two or more files based upon the keys specified (up to nine).

```
MATCH FILEA KEY1 EMPLOYEE
  FILEB KEY1 EMPNO
```

MERGE Command

Merges two or more files based upon the keys specified (up to nine).

```
MERGE FILEA KEY1 PARTNO
  FILEB KEY1 STOCK_NUMBER
AREA OUTREC
```

File or Data Print Commands

The VISION:Results file print feature contains a number of commands that you can issue in your logic to cause a record, data area, or message to be written to the print file.

The command you use depends on the nature of the data and on the type of display you want (for example, graphics, hexadecimal and graphics, hexadecimal only).

PRINT Command

Prints a file or data in graphics representation only.

```
PRINT FILEA
PRINT 'ERROR 507'
```

HEXPRINT Command

Prints a file or data in hexadecimal and graphics representation.

```
HEXPRINT FILEA
HEXPRINT FIELDA LENGTH 30
```

HEX Command

Prints a file or data in hexadecimal representation only.

```
HEX FILEA
HEX FIELDA LENGTH 30
```

LCPRINT Command

Prints a file or data in graphics representation only, but does not translate normally unprintable characters to blanks. You must use the keyword LCPRINT when you print lowercase characters.

```
LCPRINT FILEA
LCPRINT FIELDA LENGTH 30
```

REPORTFILE Command

Prints a VSE report file.

```
REPORTFILE FILEA
```

Report-Related Control Statements

This section describes the report-related control statements.

REPORT Command

Code to override VISION:Results report defaults, such as lines per page, characters per line, and carriage control. Use it also to specify a print line exit routine. Additionally, when using the multiple-report or request facility, individual requests are delineated by appending a numeric value to the REPORT command.

```
REPORT 60 LONG 84 WIDE
REPORT ASA
REPORT002 60 LONG
REPORT03
```

CONTROL or SUBTOTAL Command

Use when special processing is to be done whenever a particular field or fields change in value. This is typically called a control break. In the CONTROL statement, you code the data names from minor to major of the fields that are to be checked upon each entry to the report writer. You can use SUBTOTAL or SUBTOTAL BY (these are synonyms of CONTROL) instead of CONTROL.

```
CONTROL EMPNUM DEPT DIV
        or
SUBTOTAL BY EMPNUM DEPT DIV
```

ON CHANGE IN Command

Place ON CHANGE IN followed by one of the data names coded in the CONTROL or SUBTOTAL statement before the logic you want executed whenever that particular control break level is being processed. The ON CHANGE IN establishes an entry point.

```
CONTROL EMPNUM DEPT DIV
.
.
.
ON CHANGE IN EMPNUM
  LIST DIV DEPT EMPNUM SUM HOURS SUM SALARY
```

ON FINAL Command

Code ON FINAL preceding the logic you want to execute at the end of the report. This logic normally consists of a LIST statement to print final or grand totals.

```
ON FINAL
  LIST SUM SALARY 'GRAND TOTAL'
```

TITLE (T1-T9) Commands

You can code up to nine title or footing lines. These titles are automatically centered unless you specify otherwise. You can place variable information, such as date, page number, and user data in the titles.

```
T1 'ACCOUNTS RECEIVABLE'
T2 'MONTHLY REPORT'
T1+120 DYLDATE
```

Chapter 4: Structure of a VISION:Results Program

This chapter describes the basic structure of a VISION:Results program. It does not describe all of the features, but shows you what is involved in coding a typical program, as well as some of your options.

The following is an example of a VISION:Results program and the report output:

```
FILE ARFILE INPUT   FB 352 5280
  NAME 25 85    BALANCE 5 170 PD 2 A    ACCOUNT 2 182
FILE ARPART OUTPUT FROM ARFILE   FB 352 5280
CONTROL ACCOUNT
   IF ACCOUNT EQ 'EO' THRU 'IO' NEXT ELSE REJECT  ENDIF
  SORT ARFILE USING ACCOUNT NAME
  LIST ACCOUNT   NAME   BALANCE

ON CHANGE IN ACCOUNT
  LIST ACCOUNT   SUM BALANCE
  WITH 1 BEFORE AND 2 AFTER
ON FINAL
  LIST   SUM BALANCE
T1  'ACCOUNT EO THROUGH IO REPORT'     WITH 2 AFTER
T1+29 DYLDATE
T1+90 DYLPAGE
FIN
```

Figure 1 Basic VISION:Results Program

This program reads one input file and creates an output file containing those records with ACCOUNT EO through IO. Additionally, it creates a report that lists the selected account codes, account names, and balances owed. The balance amount is to be totaled (summed), and the total prints whenever a change occurs in the account code. Prior to listing (LIST) the fields and writing the output records, the selected records from the input file are to be sorted in NAME (minor sort key) by ACCOUNT (major sort key) sequence.

```
12/01/00            ACCOUNT EO THROUGH IO REPORT        PAGE       1

      ACCOUNT              NAME                  BALANCE

         EO        CHO PYUNG,SUH                     32.00
         EO        S.F.MEM.HOSP.                      5.00
         EO        SANTA FE HOSP ASSN                15.00

         EO                                          52.00

         FO        CANO,MICHAEL S                    15.00
         FO        CHAVEZ,RAY                        15.00
         FO        GENVARDI,G J                      43.00
         FO        HILL,GARY E                        3.80
         FO        HUGHES,RAY                       178.70
         FO        SILVA,JULIAN                       .00
         FO        TODIPE,MICHAEL                    45.24

         FO                                         300.74

         IO        AGUIERA,EMILIO                   108.44
         IO        CHAVEZ,NORMA A                    55.00
         IO        LOCKE,JEFFREY                     15.00
         IO        MONTEZ,CARMEN                     89.28
         IO        PLACIDO,ORTEGA                   413.58
         IO        VASQUEZ,IRENE                      .00

         IO                                         681.30

                                                  1,034.04
```

Figure 2 Basic VISION:Results Report

VISION:Results Program Layout

The following list is the typical organization of a VISION:Results program:

1. OPTION statements.
2. REPORT statement.
3. FILE and WORKAREA definitions.
4. Advantage™ VISION:Excel™ statements.
5. CONTROL statement.
6. If sorting, unsorted file processing followed by SORT command.
7. Detail time processing.
8. Subroutines. (Make sure detail time processing logic does not fall through into the subroutines.)
9. ON END OF SORTING.

   ON END OF INPUT.
10. ON CHANGE IN data name.

    ON FINAL.
11. TITLES and TITLE modification.

If applicable, PARM options (in VISION:Excel and VSE only) and VISION:Excel fixed-form parameters directly follow the OPTION statement. The PARM options and the FQ, SP, AG, CN, and LR fixed form parameters must be entered after the OPTION statement and before the REPORT, FILE, and WORKAREA definitions.

WORKAREAs can occur anywhere after the OPTION statement and VISION:Excel fixed form, but fields can be referenced only after they are defined.

VISION:Excel free-form statements follow FILE and WORKAREA definitions.

Comments

- An asterisk (*) in position 1 indicates a comment.
- A semicolon (;) begins a comment.
- Data that is not in the free-form columns is considered to be a comment.

OPTION Statement

Use the OPTION command to override VISION:Results defaults and supply information regarding the installation operating environment. For more information, see Program Control Commands on page 15 and the *Advantage VISION:Results for z/OS Reference Guide*.

Free-Form Columns

The free-form language defaults to columns 1 to 72. The columns that the free-form language occupies can be changed by the OPTION COLUMNS command anywhere in the program. However, the OPTION COLUMNS command must be within both the old and new free-form columns.

REPORT Statement

The REPORT statement overrides the standard VISION:Results report defaults. VISION:Results assumes a 132-character print line with 55 lines printed per page. When formatting the report columns, VISION:Results attempts to place five spaces between each printed column. You can override these defaults using the REPORT statement:

```
REPORT 85 WIDE 60 LONG 3 BETWEEN
```

Here, the report is limited to 85 characters per line instead of 132, with 60 lines per page and no more than three spaces between each column.

```
REPORT 150 WIDE
```

This report command example increases the report line to 150 characters per line. (z/OS and VSE have a maximum of 204 characters per line.)

You can place the REPORT statement anywhere in the program:

```
FILE ARFILE  ...
  NAME ...
FILE ARPART  ...
REPORT 115 WIDE
CONTROL  ...
```

FILE Statement

FILE statements provide necessary information about file characteristics and processing mode, including record format, length, block size, input or output mode, sequential or random access, and so on.

```
FILE ARFILE INPUT FB 352 5280
FILE ARPART OUTPUT FROM ARFILE  FB 352 5280
```

Every user file being processed in a program requires a FILE definition statement. For example, a program that processes an Accounts Receivable file as input and writes a partial Accounts Receivable file must have two FILE statements. The file name must immediately follow the FILE command. It can be one to eight characters if you are using z/OS and one to seven characters if you are using VSE. This file name also serves as the ddname on your z/OS JCL: it is the file name in your VSE TLBL or DLBL JCL, and the file definition name on your VM/CMS FILEDEF statement.

You could require coding more information in your FILE statement depending on the operating system and the device type. For example, under VSE (prior to VSE/SP 2.1), to define a file on 3350 disk would require:

```
FILE ARFILE INPUT FB 352 5280 DISK 3350 SYS011
```

This defines ARFILE as fixed blocked, with a record length of 352, a block size of 5280, residing on a 3350 disk, with a SYS number of SYS011. A file definition using VSE typically requires that record format, record and block sizes, and SYS number be coded. Using z/OS, file attributes can be obtained from the data set label or the JCL, so minimal information is needed in the FILE statement.

You do not need FILE statements for the files that VISION:Results knows about. These include file print, report print, sort, and work files. You need to define them in your JCL, however.

The types of files being processed in the example, a sequentially read input file and a sequentially written output file, are read and written automatically by VISION:Results during the execution. This means that prior to each entrance to your program logic, VISION:Results reads an input record (after initially opening the file) and after leaving your program, if the record has been accepted, writes a record to the output file you have defined.

When end of file is reached, VISION:Results automatically enters termination procedures, where it finishes the report, closes files, and prints I/O statistics.

In Figure 1 on page 25, VISION:Results reads an account record and enters the program logic starting with the IF statement. When the logic is finally exited (after the first LIST statement), VISION:Results writes the accepted record to the ARPART output file. The record to be written is from the input file area as specified in the 'FROM ARFILE' phrase in the FILE statement.

If you prefer to do your own reading and writing of files, you can use READ and WRITE commands in place of the automatic function in your program. For information about modifying the automatic cycle, see Modifying the Automatic Cycle on page 43.

For a complete description of the FILE statement, see the *Advantage VISION:Results for z/OS Reference Guide.*

Field Definitions

The field names appearing after the ARFILE FILE statement describe the fields in the file that are used in the program. In Figure 1 on page 25, the fields defined are to be listed in the report. Along with the name of the field, you need to give its size, its starting location (required if you are not defining all of the fields in the record), its data type if not character (NU if numeric, PD if packed decimal, or BI if binary), and its number of decimal positions if not zero.

```
NAME 25  85
```

NAME is the data name you have given to the field that is 25-bytes long, starting in location 85 of the input record. Because the data type has not been given, it is assumed to be character.

If the field is to be listed in a report, there is additional information you can put in its definition to override the VISION:Results print defaults. You can supply information about how the field is to look in a report: its printed output size, output number of decimals, edited format, and column heading.

```
BALANCE 5 170 PD 2 A
```

BALANCE has the usual information provided (size 5, location 170, type packed decimal, number of assumed decimal places 2), plus the A edit code to tell VISION:Results that if the field is listed in a report, it is to be edited with leading zero suppression, comma insertion, and zeroes printing to the right of the decimal point if the value of the field is zero. (You can also specify edit codes in the LIST statement. For a complete description, see Edit Codes on page 87.)

The value 001234567 prints as 12,345.67
The value 000000000 prints as .00

If the DYLINSTL parameter EURONUM was specified:

The value 001234567 prints as 12.345,67
The value 000000000 prints as ,00

If you do not supply an edit code, VISION:Results does not zero suppress or insert commas; it edits with a decimal point only (edit code P is the default). Negative values, under all edit formats, print with a minus sign to the right of the field (for example, 12,345.67-).

If you want to completely override the printed field defaults, code:

```
BALANCE 5 170 PD 2 A 7.0 ROUND (BALANCE OWED)
```

BALANCE is a 5-byte packed, 2-decimal field. When printed in the report, it is formatted using the A edit code, with a maximum of seven digits printing and the decimal positions eliminated. The amount is rounded before printing (for example, 500.73 would print as 501). The column heading on the report is changed from BALANCE (the default is the field name/data name as defined) to BALANCE OWED.

To save you from always having to code your field definitions in your program, you can use the VISION:Results COPY feature to put source statements such as field definitions into a library and then issue a COPY command in your program to copy them. An example of this is:

```
FILE ARFILE INPUT FB 352 5280
  COPY ARFIELDS
```

For complete descriptions of how to code field definitions and use the COPY command (including how to use COBOL data definitions in a VISION:Results program), see the *Advantage VISION:Results for z/OS Reference Guide*.

CONTROL statements typically follow the FILE and WORKAREA definitions. For more information, see CONTROL Statement/ON CHANGE IN Statement on page 35.

IF Statement

Use the IF statement to select data or certain records based on almost any criteria. You need an IF statement in those cases where you want to process only records of a certain type, or you need to execute some logic depending on conditions.

```
IF ACCOUNT EQ 'EO' THRU 'IO' NEXT ELSE REJECT ENDIF
```

Only those records with ACCOUNT codes EO through IO are selected. Any records that do not meet the criteria are rejected; they are not sorted, reported on, or written to the output file.

The keyword NEXT in the statement means that if the compare is true (ACCOUNT is equal to any value EO through IO), the next statement is to be executed. NEXT is a noise word and is never required.

Valid relational operators you can code with selection by range:

```
EQ   Equal to
NE   Not equal to
```

Some of the relational operators you can code with other formats of IF are:

```
EQ   Equal to
NE   Not equal to
LT   Less than
LE   Less than or equal to
GT   Greater than
GE   Greater than or equal to
```

You can also test bits and test for numeric, negative, or positive. You can compare a data name to another data name, a numeric literal, or a character literal:

```
IF ACCOUNT GT ACCOUNT2  ...
IF ACCOUNT EQ 'KO'  ...
IF ACCOUNT EQ 'KO' 'RS' 'QE'  ...
IF ACCOUNT EQ 'KO' ACCOUNT2  ...
IF BALANCE GT PREVBAL  ...
IF BALANCE LE 0  ...
IF BALANCE GT 500.50  ...
```

Character literals are expressed with quotation marks around them, but quotation marks are not required for numeric literals.

You may occasionally need to test a field that cannot be expressed in the normal manner. Perhaps you have to see if a field contains all high-values. You would then code a hexadecimal literal:

```
IF ACCOUNT EQ X'FFFF' ...
```

The 2-byte character field ACCOUNT is tested to see if it is hexadecimal Fs (high-values). To express the literal, start with an X' followed by two characters for every byte you are testing. The value of each character can be in the hexadecimal range of 0-F. The literal is terminated by another quotation mark.

The IF statement has considerably more versatility than what is illustrated in the example. You can express compound IF statements, do series or range testing, and branch backward and forward (GOTO):

```
IF ACCOUNT EQ 'IO' AND NAME EQ 'MARTIN'
   GOTO TAG3 ENDIF
IF ACCOUNT EQ 'EO' 'JA' 'RT' ACCEPT
   ELSE REJECT ENDIF

IF BALANCE EQ 500 THRU 999.99 NEXT ELSE GOTO TAG7 ENDIF
```

For more information on the IF statement, see the *Advantage VISION:Results for z/OS Reference Guide.*

SORT Statement

Use the SORT statement to sort the input file records in the sequence you require before the records are reported, printed, or written to an output file.

The records being sorted can be from sequential (fixed, variable, or spanned) files, VSAM files, or database files, as well as records built in a work area in your program.

You can code your SORT statement at the very beginning of your program to sort all input records, or you can select certain records or build work area records before sorting.

In the following example, selection is performed before the sort because not all of the records are to be reported on or written. This prevents you from having to sort the entire file.

```
IF ACCOUNT EQ 'EO' THRU 'IO' NEXT ELSE REJECT ENDIF
SORT ARFILE USING ACCOUNT NAME
```

Follow the SORT command by the file name to specify where to find the record to be sorted. Following the file name is the keyword USING, which always precedes the names of the fields you are sorting on (the sort key fields). You code these fields from major to minor. ACCOUNT is the major key, and NAME is the minor key. The records are then put in ACCOUNT sequence (for example, EO, EO, EO, EP, EP) with the NAMEs within the account code also in sequence.

There are a number of variations to the basic sort, including sorting only part of the record, sorting from a work area, and sorting on descending keys.

If you are operating under VSE and want to use more than one sort work file, you can specify how many sort work files you require in the SORT statement:

```
SORT ARFILE USING ACCOUNT NAME WORK 3
```

For a complete description of the SORT command, see the *Advantage VISION:Results for z/OS Reference Guide*.

FIN Statement

The FIN statement concludes the VISION:Results program statements. It is required only when an instream data file is to follow the program. The previous example does not require the use of FIN.

LIST Statement

The LIST statement describes what fields are to be printed in the report, how they are to be edited, and in what order they are to be printed from left to right. The fields you are listing do not have to be from an input file—they can also be from a work area, or they can be literals.

VISION:Results automatically formats the report so that the fields are properly positioned on the page and are printed in the order specified.

```
LIST ACCOUNT NAME BALANCE
```

In the example, the fields ACCOUNT, NAME, and BALANCE are printed.

Column headings are handled automatically. The width of each column is determined by whichever is larger:

- The column heading of the field printed.
- The field size as defined.
- The edit mask used by VISION:Results, if the field is NU, BI, or PD.

VISION:Results defaults to using the data names as the column headings (for example, ACCOUNT, NAME, BALANCE). You can override this in the field definition:

```
ACCOUNT 2 182 (ACCOUNT'CODE)
```

In addition, you can override data names as column headings in the LIST statement itself:

```
LIST ACCOUNT (ACCOUNT'CODE) ...
```

Indexed data names can be entered in the LIST statement. For example:

```
FILE FILE FB 80
  RECORD1 80
  FIELD1 5 1
  FIELD2 5
  FIELD3 5

INX - 0

LIST FIELD1 (INX) (HEADER1) FIELD2

FIN
```

Figure 3 Example of Indexed Data Names

If you do not want headings to appear over a particular column, code:

```
LIST ACCOUNT ()
```

What you code in the LIST statement for a column heading overrides what is coded in the field definition.

You can specify literals to print in your LIST statement:

```
LIST ACCOUNT NAME BALANCE  '--'
```

If you code a literal in your LIST statement, you cannot give it a column heading.

You can list a literal that also has a column heading by defining the literal with a column heading in a work area and using its data name in the LIST statement.

```
WORKAREA

   LITFIELD 2 ('CHECK'OFF') VALUE '--'
   LIST ACCOUNT NAME BALANCE LITFIELD
```

```
12/01/00          ACCOUNT BO THROUGH IO REPORT       PAGE       1

      ACCOUNT            NAME               BALANCE       CHECK
                                                            OFF
        BO       CHO PYUNG,SUH                  32.00      --
        BO       S.F.MEM.HOSP.                   5.00      --
        BO       SANTA FE HOSP ASSN             15.00      --

        BO                                      52.00
```

Figure 4 Example of List Account Name Balance

You can do multi-line printing on either detail or control break lines, up to 99 lines:

```
FILE NAMEIN  ...
 COPY NAMEDEFS
REPORT 35 WIDE 3 BETWEEN
```

```
LIST NAME ()
LIST ADDRESS ()
LIST CITY () STATE ()  ZIP ()
     WITH 2 AFTER
```

VISION:Results creates a three-line report, with ADDRESS lined up under NAME, and CITY lined up under ADDRESS.

```
JOHN ABRAMSON
123 EVERGREEN ST
LOS ANGELES      CA   90034

LILAH D. PHILIPS
4043 MARTIN BLVD
ROLLING HILLS    CA   91134
```

On a multi-line report (accomplished by multiple LIST statements), if you are having VISION:Results determine where the fields are to be placed on the line, VISION:Results normally tries to align a field to a field in the previous LIST statement that is in the same relative position. In the name/address example, the Line 2 ADDRESS field is aligned with the Line 1 NAME field because they are the first fields to be listed. Accordingly, CITY is aligned under ADDRESS. STATE and ZIP have nothing to line up with, so they are given their own columns.

You can also control where fields are printed on the line. In addition, you can position a field by aligning it relative to a previous field on that line or a previous line.

```
LIST CITY ()
LIST STATE AT CITY + 20   ZIP AT CITY + 25
```

To have complete control over where VISION:Results positions fields, code absolute print locations.

```
LIST  NAME AT 1
LIST  ADDRESS AT 1
LIST  CITY AT 1   STATE AT 21   ZIP AT 25
```

If you use the absolute print location option, you must specify it for every field you are printing. You cannot mix absolute with relative or automatically composed positioning, but you can mix relative and composed positioning. You do not get column headings if you use absolute positioning. You can still specify headings and footings by the 'Tn' statements and provide column headings in these statements.

VISION:Results defaults to single-spacing when printing report lines. The line space control defaults are zero lines before printing a line and one line after printing. You can easily specify any line spacing in 1-line increments.

```
LIST ACCOUNT SUM BALANCE WITH 1 BEFORE AND 2 AFTER
```

The phrase WITH 1 BEFORE AND 2 AFTER causes a blank line to occur before the line is printed and an extra blank line to occur after the line is printed. You can do this when printing subtotal (control break) lines to make them more identifiable in the report.

In multi-line reporting, on other than the first LIST statement, you can only specify AFTER spacing. You can only state spacing BEFORE in the first LIST statement in a series.

Another useful feature during report printing is to be able to start a new page after a specific event has occurred, such as when the ACCOUNT changes from one value to another or after printing the subtotals for the previous ACCOUNT records.

```
LIST ACCOUNT SUM BALANCE WITH 1 BEFORE AND EJECT AFTER
```

CONTROL Statement/ON CHANGE IN Statement

The printing of detail lines in a report has been described so far, with brief references to another important part of report printing—subtotal or control break lines. The LIST ACCOUNT NAME BALANCE statement in the previous example is all you need to list the desired fields of every record you selected. You can also run a total on BALANCE and print this subtotal along with its corresponding Account Code whenever the Account Code changes in value. In other words, you can print a total of all the EO account balances, FO account balances, and so on. In the example, this is what happens after each of the ACCOUNT series: the 52.00 balance subtotal for Account Code EO, the 300.74 subtotal for FO, and the 681.30 for IO.

To tell VISION:Results what fields need to be checked for a change in value to cause a control break, and print lines or execute logic at that certain time, use a CONTROL statement and an ON CHANGE statement as follows:

```
CONTROL ACCOUNT
.
.
ON CHANGE IN ACCOUNT
  LIST ACCOUNT  SUM BALANCE
```

The CONTROL statement tells VISION:Results which fields to check for a change in value. You must define all of the fields before coding them in a CONTROL statement, and place the CONTROL statement before any ON CHANGE IN statements. In the example, the ACCOUNT field is checked for a change before listing a detail line. If there is a change (for example, the value in the record has gone from EO to EP), the logic coded after the ON CHANGE IN ACCOUNT is executed. Usually, you have a LIST statement after the ON CHANGE IN specifying the printing of the control break field and the subtotals of amount fields.

Additionally, you can execute other statements such as compute formula, build and write a summary record, test a total field:

```
ON CHANGE IN ACCOUNT
  IF SUM BALANCE GT 75000.00 MOVE '*' TO COMMENT ENDIF
    LIST ACCOUNT SUM BALANCE COMMENT
```

Here, at ACCOUNT control break time, the ACCOUNT balance subtotal is checked to determine if it is over a certain amount. If so, the control break line prints with an asterisk appearing to the right of the BALANCE column. (Assume that the COMMENT field has been defined in a work area earlier in the program coding.)

Note: You can state the keyword SUM, which precedes BALANCE, during any of the ON CHANGE IN or ON FINAL logic to indicate that a subtotal or final total at that particular level is to be used instead of a BALANCE amount from a single record. For example, after ON CHANGE IN ACCOUNT, LIST ... SUM BALANCE was coded. In other words, list the total balance amount of all records within that particular account code.

You could also code:

```
MOVE SUM BALANCE TO XYZ
```

or

```
AVERAGE = SUM BALANCE/TALLY
```

To add some extra line spacing at control break time and not print fields, you could code the following to create blank lines between one account code detail series and the next:

```
ON CHANGE IN ACCOUNT
  LIST ' ' WITH 1 BEFORE
```

To cause a page eject between each account code detail series:

```
ON CHANGE IN ACCOUNT
  LIST ' ' WITH EJECT AFTER
```

If you are issuing a LIST after the ON CHANGE IN or ON FINAL logic, make it the last statement in that section. It is always executed last.

You can code up to six fields (data names) in the CONTROL statement, starting with minor and proceeding to major (not including grand total).

In this accounts receivable example, assume you also wanted to do something special when the NAME within the ACCOUNT changed, as well as when the ACCOUNT changed. In the sort, SORT ARFILE USING ACCOUNT NAME is already set up to do this. The names are in sequence by Account Code. If you had only sorted on account code (SORT ARFILE USING ACCOUNT), breaking on NAME within ACCOUNT would not work because the same-named accounts within the particular account code would not necessarily be grouped together.

```
CONTROL NAME ACCOUNT
   .
   .
SORT ARFILE USING ACCOUNT NAME
   .
   .
ON CHANGE IN NAME
  LIST NAME SUM BALANCE WITH 1 BEFORE AND 2 AFTER
ON CHANGE IN ACCOUNT
  LIST ACCOUNT SUM BALANCE WITH 3 AFTER
```

Example

```
12/01/00       ACCOUNT EO THROUGH IO REPORT      PAGE   1

      ACCOUNT              NAME             BALANCE

        EO          CHO PYUNG,SUH              32.00
```

Figure 5 Example of List Account Sum Balance Report (Page 1 of 2)

```
                CHO PYUNG,SUH              32.00
        EO      S.F.MEM.HOSP.               5.00
                S.F.MEM.HOSP.               5.00
        EO      SANTA FE HOSP ASSN         15.00
                                           15.00
        EO                                 52.00
        FO      CANO,MICHAEL S             15.00
```

Figure 5 Example of List Account Sum Balance Report (Page 2 of 2)

ACCOUNT is the major control break (second break). This is indicated in the CONTROL statement by coding it last. It is the field that VISION:Results tests first to see if a control break has occurred. If the ACCOUNT value has not changed, VISION:Results checks the next lower control break field, NAME.

If VISION:Results does find that the ACCOUNT value has changed, it first does what has been specified in the minor break (break 1) ON CHANGE IN NAME logic. In this case, it lists a line showing the ACCOUNT and the BALANCE subtotal for the account name within that account code.

```
SANTA FE HOSP ASSN             15.00
```

After executing the minor control break logic, VISION:Results executes what has been specified after the ON CHANGE IN ACCOUNT major control break logic. Now, as directed in the LIST statement, it lists only the ACCOUNT and the subtotal for all of the balances within the account code.

```
EO                              52.00
```

If VISION:Results finds that only a change in NAME has occurred, it executes the logic following the ON CHANGE IN NAME statement.

ON FINAL Statement

Often, at the end of a run when all records have been reported, you want to print overall totals or other information. In the accounts receivable example, at the bottom of the report there is a grand total of all balances owed by all of the selected account codes.

To perform logic or to list fields at grand total time requires the following:

```
ON FINAL
 .
 .  (your statements here)
 .
```

The ON FINAL statement precedes logic you want performed before the report is finished.

```
ON FINAL
  LIST  SUM BALANCE
```

Here, the only thing that prints is a sum of the balances. It is not appropriate to print NAME or ACCOUNT at this time because what you want printed applies to all account codes and all names.

You do not need to code anything in the CONTROL statement to indicate a grand total level. This action is not considered a control break.

Again, you are not restricted to only LIST statements. You can test total fields, move fields, or branch to other logic.

Report Titles and Footings (Tn)

You have seen how column headings are specified by the data name, in the field definition, or on the LIST statement. To specify report headings or footings requires a T1 through T9 statement, followed by the text enclosed in single quotation marks.

```
T1 'ACCOUNT EO THROUGH IO REPORT'
```

You can use a 1- to n-character title. However, if the title is wider than 70 characters, the title specification must be completed on the next line.

```
T1 'xxxx xxxxxx xxxxx xxxx xx xxxxxxx xxxx xxxxx xx xxxx xxxxx xxxxx xxxxxx'
T1+70 'xxxxxxxxxxxxxxxxxxxxxxxxxx'
```

Whenever a page eject occurs in the report, VISION:Results automatically prints the title centered on the page, followed by the column headings, then a report line.

```
12/01/00          ACCOUNT EO THROUGH IO REPORT       PAGE       1

      ACCOUNT             NAME                     BALANCE

         EO       CHO PYUNG,SUH                       32.00
         EO       S.F.MEM.HOSP.                        5.00
         EO       SANTA FE HOSP ASSN                  15.00

         EO                                           52.00

         FO       CANO,MICHAEL S                      15.00
         FO       CHAVEZ,RAY                          15.00
         FO       GENVARDI,G J                        43.00
         FO       HILL,GARY E                          3.80
         FO       HUGHES,RAY                         178.70
         FO       SILVA,JULIAN                          .00
         FO       TODIPE,MICHAEL                      45.24

         FO                                          300.74

         IO       AGUIERA,EMILIO                     108.44
         IO       CHAVEZ,NORMA A                      55.00
         IO       LOCKE,JEFFREY                       15.00
         IO       MONIEZ,CARMEN                       89.28
         IO       PLACIDO,ORTEGA                     413.58
         IO       VASQUEZ,IRENE                         .00

         IO                                          681.30

                                                   1,034.04
```

Figure 6 Example of Title Centered on the Page

Code the following if you want footings to appear at the bottom of the page:

```
T2 'REPORT 1012-7' FOOTING
```

On the bottom of each page of the report, the literal 'REPORT 1012-7' is centered and printed.

You can have up to nine Tn statements (T1 through T9) with any combination of headings and footings.

```
T1 'ACCOUNT EO THROUGH IO REPORT'
T2 'MONTHLY'
T3 'REPORT 1012-7' FOOTING
```

You can include variable information, such as date, page number, and account code, into your headings or footings. This is done in the following manner:

```
T1+29 DYLDATE
T1+90 DYLPAGE
```

DYLDATE and DYLPAGE are special VISION:Results names (called reserved words) that identify fields you want to print in your titles:

- DYLDATE is the current date in edited format mm/dd/yy.
- DYLPAGE is the report page number that is maintained by VISION:Results. It has the format PAGE nnnnnnn.

In the example, DYLDATE is placed at position 29 and DYLPAGE is at position 90 of the title. For example:

```
12/01/00       ACCOUNT EO THROUGH IO REPORT      PAGE        1
```

Figure 7 Example of DYLDATE Placement

For a complete list of reserved words that you can place in your titles or reference in your program, see the *Advantage VISION:Results for z/OS Reference Guide*.

Additionally, you can enter your own data into titles. For example, instead of the page number appearing at the upper right of each page, you may want to print the Account Code being reported on at that time:

```
T1+101 ACCOUNT
```

If you are entering this kind of information in the title, you want to eject after listing the ON CHANGE IN ACCOUNT control break line so you never have different account codes on the same page.

You can code the Tn+n statements only if you have a corresponding Tn 'xxx' statement. You can also specify how many blank lines are to appear between each title and the column headings. The default is single spacing, that is, no blank lines. Typically, you want a blank line to appear between your last title line and your column headings, as shown in the following:

```
T1 'ACCOUNT EO THROUGH IO REPORT' WITH 2 AFTER
```

In the WITH n AFTER statement, you can specify 1 through 9 for the n range. The default value is 1.

Summary Report

There are times when you only want to produce a summary report. In this instance, you are not interested in having information on each record printed, only a recap or summary of groups of records.

In the accounts receivable example (Basic VISION:Results Program on page 25 and Basic VISION:Results Report on page 26), if you only wanted to list the subtotals of the balances owed by each account code group and the overall balances owed, you would code:

```
FILE ARFILE INPUT
  ACCOUNT 2 182  BALANCE 5 170 PD 2
CONTROL ACCOUNT
IF ACCOUNT EQ 'EO' THRU 'IO' NEXT ELSE REJECT ENDIF
SORT ARFILE USING ACCOUNT
ON CHANGE IN ACCOUNT
  LIST ACCOUNT SUM BALANCE
ON FINAL
  LIST SUM BALANCE WITH 2 BEFORE
T1 'SUMMARY REPORT'
```

Figure 8 Example of a Summary Report

The difference between a detail plus summary report and a summary only report is the absence of the detail LIST statement.

```
            SUMMARY REPORT

      ACCOUNT           BALANCE

         EO               52.00
         FO              300.74
         IO              681.30

                        1,034.04
```

Figure 9 Example of the Report of a Summary Report

If you only wanted to print grand totals, the program would look like:

```
FILE ARFILE  INPUT
  ACCOUNT 2 182  BALANCE 5 170 PD 2
CONTROL ACCOUNT
IF ACCOUNT EQ 'EO' THRU 'IO' NEXT ELSE REJECT   ENDIF
SORT ARFILE USING ACCOUNT
ON FINAL   LIST SUM BALANCE
```

Figure 10 Example of Printing Grand Totals

VISION:Results Automatic Cycle

The VISION:Results automatic cycle executes your program in the following sequence of events.

First, VISION:Results reads and validates your program statements. If nothing is in error, it generates the code necessary to perform the functions you have specified. Next, execution of the program begins.

In program execution, VISION:Results first opens the file and then reads a record from any sequentially accessed input file you have defined by a FILE statement. It then processes your program statements where it selects records, reformats records, or reports on fields. When there are no more statements to execute, an ON CHANGE IN has been encountered, or you have ACCEPT or REJECT in your statements, VISION:Results exits the program.

When the statements are exited, the output part of the cycle begins. If you indicated in an OPTION statement that you want to do a file print of a particular file (for example, OPTION PRINT ARFILE), a record is printed if you have not rejected (REJECT). After this, any output files you have defined by the FILE statement are written. After all output specifications have been performed, VISION:Results begins another input cycle.

The following example is a typical program that shows the automatic cycle.

```
FILE ARFILE INPUT FB 352 5280
  NAME 25 85  BALANCE 5 170 PD 2 A  ACCOUNT 2 182
FILE ARPART OUTPUT FROM ARFILE FB 352 5280
CONTROL ACCOUNT
  IF ACCOUNT EQ 'EO' THRU 'KO' NEXT ELSE REJECT ENDIF
  LIST ACCOUNT NAME BALANCE
ON CHANGE IN ACCOUNT
  LIST ACCOUNT SUM BALANCE
ON FINAL
  LIST SUM BALANCE
```

Figure 11 Example of a Typical Automatic Cycle Program

In this program, VISION:Results reads the accounts receivable file, selects records in the account code range of EO through KO, reports on these selected records, and writes the selected records to a sequential output file. A simplified flowchart of what occurs is shown in Figure 12.

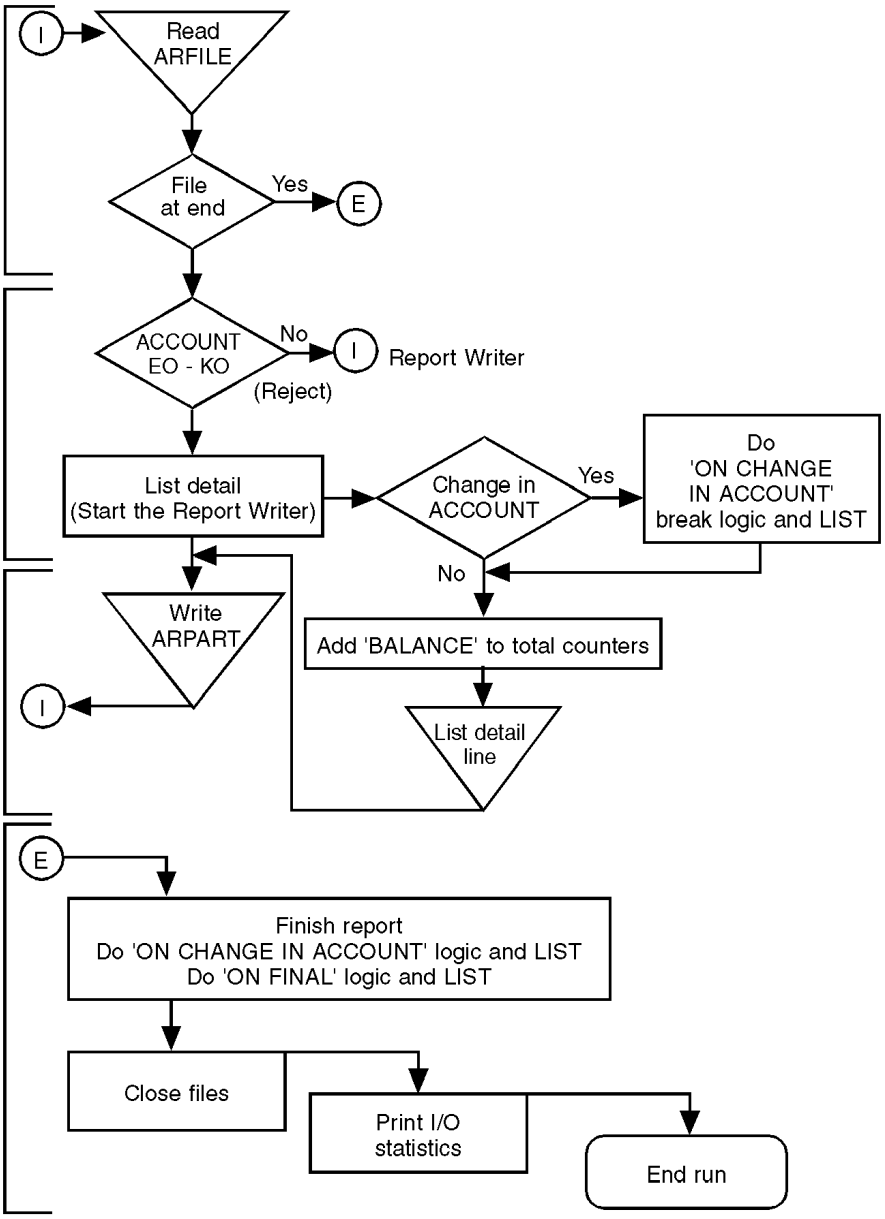

Figure 12 The Automatic Cycle

INPUT phase	VISION:Results reads an input record and, if it is not at end of file, passes control to the PROCESS phase.
PROCESS phase	VISION:Results executes the user's program logic statements and tests the account code for the range between EO and KO: ■ If the record is within this range, the next statement (LIST) is executed, which causes the report writer module to be entered to print a line on the report. ■ If the record is not within the range that you want (between EO and KO), it is rejected, causing all output functions to be bypassed, which in this case, means the record is not written to the output file. After VISION:Results finishes the report (LIST) functions, it exits the user's program logic statements by an implied ACCEPT. (The ON CHANGE IN statement has been reached and is not executed at this time. It is executed by the report generator when it has detected a control break.)
OUTPUT phase	VISION:Results writes any accepted records to the output file as defined in the FILE statement. After this, the cycle begins again.
END OF RUN	After VISION:Results reads the file completely, termination procedures begin. The report writer module is entered, where it executes the ON CHANGE IN ACCOUNT logic to print the last control break line and then completes the report by executing the ON FINAL logic to print the grand totals. VISION:Results then closes all active files, prints run statistics, and ends the run.

Modifying the Automatic Cycle

The automatic cycle of VISION:Results means you can code a program quickly and error free. Also, for many program requirements, the I/O sequence fits in with the automatic cycle. However, you are not restricted to having it controlled for you. You can override any of these automatic events so that you can read, write, and print at any time and in any sequence.

You can include the following commands in your program statements to bypass the automatic activity of VISION:Results in the input or output phase of the cycle so that you control it.

- READ file name:
  - Read a record from the file specified.
    Example: `READ ARFILE`
- WRITE file name:
  - Write a record to the file specified.
    Example: `WRITE ARPART`

To print a record, field, or literal to the SYSPRINT (z/OS and VM/CMS), or SYSLST (VSE) file, use:

```
{HEXPRINT | PRINT | LCPRINT | HEX}   {filename} [length {dataname}]
                                     {dataname} [       {n       }]
                                     {literal }
```

where:

HEXPRINT	Prints the record or field in both hexadecimal and graphics representation.
PRINT	Prints the record or field in graphics only representation.
LCPRINT	Prints the record or field in graphics only representation without translation of unprintable characters. Specify when printing data containing lowercase characters.
HEX	Prints the record or field in hexadecimal representation only.

Examples:

```
PRINT ARFILE                (Prints a record from the file ARFILE.)
HEXPRINT BALANCE            (Prints the field called BALANCE.)
PRINT '**ERROR 21**'        (Prints the message literal.)
```

In Figure 13, the previous program has been changed to control the input and output functions in the user's program logic statements rather than have VISION:Results do this as part of the automatic cycle.

```
FILE ARFILE INPUT FB 352 5280
NAME 25 85  BALANCE 5 170 PD 2 A  ACCOUNT 2 182
FILE ARPART OUTPUT FROM ARFILE FB 352 5280
CONTROL ACCOUNT
  READ ARFILE
  IF ACCOUNT EQ 'EO' THRU 'KO' NEXT ELSE REJECT ENDIF
  LIST ACCOUNT NAME BALANCE
   WRITE ARPART
ON CHANGE IN ACCOUNT
  LIST ACCOUNT SUM BALANCE
ON FINAL
  LIST SUM BALANCE
```

Figure 13 Example of Controlling the Input and Output Functions

The logic takes control of the reading of the ARFILE and the writing of ARPART. In this example, the logic does not do anything different than if VISION:Results does the I/O automatically, but it provides more complete documentation in the program as to what functions are occurring.

Your control of I/O functions usually becomes necessary when you are processing more than one input file or where processing of input and output is conditional.

If you are reading random VSAM files, or reading VSAM in skip sequential mode, you have to issue reads within your program. This is because VISION:Results does not know which record to read or write, or where to start reading, until you code it in the program by supplying a full or generic key.

If you are writing a random VSAM file, you must first issue the READ for a particular record and later issue a WRITE of the record.

The following program reads the accounts receivable master file and uses the 2-byte account code (ACCOUNT) in the record to retrieve an account code description in a keyed sequence VSAM file. the program lists this description in the report along with the account code.

```
FILE ARFILE INPUT FB 352 5280
   NAME 25 85  BALANCE 5 170 PD 2 A   ACCOUNT 2 182
 FILE ACCTDESC KSDS F RANDOM FINDKEY
   KEYLEN 2    STATUS STATUSFLAG
*FIELD DEFINITIONS FOR ACCTDESC
   VACCOUNT 2 1   VDESCRIPT 15 3 (ACCOUNT CODE'DESCRIPTION)
CONTROL ACCOUNT
READAR:
 READ ARFILE
 IF ARFILE EQ X'FF' GOTO ENDRUN ENDIF
 MOVE ACCOUNT TO FINDKEY
 READ ACCTDESC
 IF STATUSFLAG NE 'Y' GOTO ERROR1 ENDIF
 LIST ACCOUNT   VDESCRIPT   BALANCE
 GOTO READAR
ENDRUN:
 STOP
ERROR1:
   PRINT 'RECORD NOT FOUND'
   PRINT ACCOUNT
   GOTO ENDRUN
ON CHANGE IN ACCOUNT
 LIST ACCOUNT   SUM BALANCE
      WITH 2 BEFORE AND 3 AFTER
ON FINAL
 LIST SUM BALANCE
```

Figure 14 Example of a Program That Reads the Accounts Receivable File

Note that after the READ ARFILE command, there is a test to find out if the file is at end. When VISION:Results finds that a sequentially read file has reached the end, it places high-values in the record area of ARFILE. You can then test for this condition by comparing ARFILE (actually the first byte of the accounts receivable file) to X'FF'. (Besides setting the status indicator, VISION:Results completely fills the file record area with high-values (X'FF's).) When this is true, the program branches to the tag name ENDRUN, where the program is terminated. The reason you are terminating the run yourself is that the VSAM file is still active and VISION:Results does not automatically terminate the run for you.

Any time you want to terminate execution of your program before VISION:Results does, issue a STOP command. When you do this, VISION:Results finishes the report by performing all of the ON CHANGE IN and ON FINAL logic from minor to major. It then closes the files, prints the run statistics, and terminates.

In addition to the report (LIST), the file print (PRINT) feature is being invoked to print an error message the first time an account code description record cannot be found on the VSAM file. Under z/OS and VM/CMS, the message goes to the SYSPRINT file; under VSE, it goes to SYSLST. If you are operating under VSE, be careful how you code your LISTs and PRINTs. The report defaults to SYSLST and you find record printouts or messages interspersed with your report.

The error message looks something like:

```
RECORD NOT FOUND
EO
```

Modes of Operation

VISION:Results provides three programming modes, which you can select based on your requirements. Once you select a mode, it remains in effect until you change it. Indicate the programming mode by the OPTION statement.

The three modes are:

- Conventional mode—This is the default and it sets the mode to standard VISION:Results. Unless specified otherwise, the information and examples in this guide refer to programming in the conventional mode.
- Structured mode—This mode and its variation, Structured2, allow use of all structured programming features in VISION:Results. For a complete discussion of this option, see Chapter 6: Structured and USERDEFAULT Mode Programming.
- USERDEFAULT—The USERDEFAULT mode assumes certain defaults to simplify use of VISION:Results. For the novice programmer, the USERDEFAULT mode is an easy way to produce lists and reports. For a complete discussion of this option, see USERDEFAULT Mode Programming on page 80.

Chapter 5: Programming Examples

This chapter describes how to solve problems by using VISION:Results.

A step-by-step approach has been taken with the first 11 examples, with new commands or keywords being added to the initial solution. The next four examples introduce additional commands and keywords, as well as demonstrate the flexibility and versatility of VISION:Results. The remaining five examples are a few of the utility functions available in VISION:Results. No pattern has been established here—each example is unique. The FILE statements in the examples that follow reflect z/OS and VM keyword requirements. For VSE requirements for FILE statements, see the *Advantage VISION:Results for z/OS Reference Guide*.

The list below includes some of the automatic features covered in the examples:

- Centering of headings and titles
- Using field names as column headings
- Automatic centering of data under column headings
- Editing of fields
- Determining control breaks and printing control break lines
- Totaling fields
- Initializing and re-initializing
- Rounding

The examples illustrated in this chapter are a small sample of what you can do with VISION:Results.

- For information about programming in the Structured and USERDEFAULT modes, see Chapter 6: Structured and USERDEFAULT Mode Programming and USERDEFAULT Mode Programming on page 80.
- For additional commands and keywords, see the *Advantage VISION:Results for z/OS Reference Guide*.

Sorting and Printing Details for Each Field

Generate a report in account code sequence with column information on the account code, transaction number, name, and balance. (For the definition of the statements, commands, and keywords, see Commands, Keywords, and Definitions on page 81.)

```
FILE ARFILE  INPUT  FB   352   5280
    ACCOUNT 2 182   NAME 25 85   TRANS 7 4
    BALANCE 5 170 PD 2 A
SORT ARFILE USING ACCOUNT
LIST ACCOUNT TRANS NAME BALANCE
T1 'VISION:RESULTS REPORT 1'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 15 Sort and Print Detail for Each Field

The statements above generated the report in Figure 16 on page 48. The data was first sorted into account code sequence, and then reported on. The keyword INPUT is the default value for the command FILE and is not used after this example.

Each field definition used is given a data name, size (length), position in the input record, data format, number of assumed decimal positions (if required), and edit mask (if wanted). Library facilities are available for field definitions using the COPY function. For example purposes, however, the field definitions are included in the program, except for Example 4.

```
12/01/00              VISION:RESULTS REPORT 1             PAGE 1

  ACCOUNT     TRANS            NAME                 BALANCE

     BO      8006547   TORRES,ERNESTO                   44.99
     EO      6208657   CHO PYUNG,SUH                    32.00
     EO      6002587   SANTA FE HOSP ASSN               15.00
     EO      7082509   S.F.MEM.HOSP.                     5.00
     FO      8011508   HUGHES,RAY                      178.70
     FO      6024963   HILL,GARY E                       3.80
     FO      6044395   CANO,MICHAEL S                   15.00
     FO      6059708   CHAVEZ,RAY                       15.00
     FO      6123228   SILVA,JULIAN                      .00
     FO      6095631   TODIPE,MICHAEL                   45.24
     FO      6107265   GENVARDI,G J                     43.00
     IO      2002299   PLACIDO,ORTEGA                  413.58
     IO      6218113   AGUIERA,EMILIO                  108.44
     IO      8012644   MONTEZ,CARMEN                    89.28
     IO      6009166   LOCKE,JEFFREY                    15.00
     IO      6112536   CHAVEZ,NORMA A                   55.00
     IO      6123317   VASQUEZ,IRENE                     .00
     KO      6004695   SANTA FE RR ASSN                 32.02
     KO      2005131   WICINSKI,ALEXANDER B            848.71
     KO      6016316   VANCE,VERNON                     40.76
     KO      6062946   BROCK,EITHEL G                   77.00
     KO      6067956   PORTER,MAY T                     80.24
     KO      6049354   SWARTZ,GEORGE                    21.25
     KO      6041272   MARTIN,GAYLORD L                 15.00
     KO      6042325   CHAMBERLINE,FRANCES              34.24
     KO      1014021   WOOD, ROGER D                 1,702.42
     KO      6088678   TYLER,JAMES                     305.64
     KO      6080669   WILLIAMS,JAMES                   24.24
     KO      8033692   MARLETTE,YVETTE                  14.95
     KO      9005226   WILSON, W C                      22.40
     KO      7100035   BROWN, RICHARD                   50.62
     KO      6216129   MUNSON,CLARENCE                  15.00
     KO      6218512   MARTIN,GAYLORD L                 55.25
     KO      7029926   WHITE, HIAWATHA                  20.44
     KO      7014996   OLVERA, CLEMENTINE              106.74
     KO      7072694   WARTON, LEE                     224.00
```

Figure 16 Report in Account Code Sequence (Page 1 of 2)

```
KO       6116728    STEINER,ROBERT                15.00
KO       6101291    RACH,MARY                     35.00
KO       6103375    GARKOW,DOROTHY                25.00
LO       6114113    NERY,GENEROSO                 45.24
MA       9017828    HILL, MYRTLE G                11.20
MA       7106246    SMITH, AUSTIN                  5.12
MA       7102194    TURNER, HAROLD                 5.12
MA       7099657    MALITSBERGER, JOHN A           6.72
MA       7023189    SIAS, EDWARDO                   .00
MA       7033133    DE VAULT, JERRY               11.20
MA       7011407    WHITE, ELMER                  10.24
MA       7039085    BLACK, LENORE                 50.40
MA       7010966    SMITH, JOHN                   13.66
MA       7012799    RODRIGUEZ, EVARISTO           10.58
MA       7004753    FORBES, JOHN                  11.20
```

Figure 16 Report in Account Code Sequence (Page 2 of 2)

Printing Selected Field when Value Changes

In this example, the field containing the account code is printed only when the value of that field has changed. Additionally, the report title is modified to identify each example.

```
FILE ARFILE FB 352 5280
  ACCOUNT 2 182   NAME 25 85   TRANS 7 4
  BALANCE 5 170 PD 2 A
SORT ARFILE USING ACCOUNT
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE
T1 'VISION:RESULTS REPORT 2'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 17 Print Selected Field when Value Changes

```
12/01/00              VISION:RESULTS REPORT 2            PAGE 1

  ACCOUNT     TRANS          NAME                    BALANCE

    BO       8006547    TORRES,ERNESTO                 44.99
    EO       6208657    CHO PYUNG,SUH                  32.00
             6002587    SANTA FE HOSP ASSN             15.00
             7082509    S.F.MEM.HOSP.                   5.00
    FO       6059708    CHAVEZ,RAY                     15.00
             6123228    SILVA,JULIAN                    .00
             6095631    TODIPE,MICHAEL                 45.24
             6107265    GENVARDI,G J                   43.00
             8011508    HUGHES,RAY                    178.70
             6024963    HILL,GARY E                     3.80
             6044395    CANO,MICHAEL S                 15.00
    IO       2002299    PLACIDO,ORTEGA                413.58
             8012644    MONTEZ,CARMEN                  89.28
             6009166    LOCKE,JEFFREY                  15.00
             6112536    CHAVEZ,NORMA A                 55.00
             6123317    VASQUEZ,IRENE                   .00
             6218113    AGUIERA,EMILIO                108.44
    KO       6004695    SANTA FE RR ASSN               32.02
             2005131    WICINSKI,ALEXANDER B          848.71
             6016316    VANCE,VERNON                   40.76
             6042325    CHAMBERLINE,FRANCES            34.24
             1014021    WOOD, ROGER D                1,702.42
             6088678    TYLER,JAMES                   305.64
             6080669    WILLIAMS,JAMES                 24.24
             8033692    MARLETTE,YVETTE                14.95
             9005226    WILSON, W C                    22.40
             7100035    BROWN, RICHARD                 50.62
             6216129    MUNSON,CLARENCE                15.00
             6218512    MARTIN,GAYLORD L               55.25
```

Figure 18 Report with Account Codes Suppressed Between Changes (Page 1 of 2)

	7029926	WHITE, HIAWATHA	20.44
	7014996	OLVERA, CLEMENTINE	106.74
	7072694	WARTON, LEE	224.00
	6116728	STEINER,ROBERT	15.00
	6101291	RACH,MARY	35.00
	6103375	GARKOW,DOROTHY	25.00
	6062946	BROCK,ETHEL G	77.00
	6067956	PORTER,MAY T	80.24
	6049354	SWARTZ,GEORGE	21.25
	6041272	MARTIN,GAYLORD L	15.00
LO	6114113	NERY,GENEROSO	45.24
MA	7010966	SMITH, JOHN	13.66
	7012799	RODRIGUEZ, EVARISTO	10.58
	7004753	FORBES, JOHN	11.20
	7012837	WOODS, LOUISE	.00
	6216412	ROUKE,CURTIS	48.00
	6219004	SCOLEY,WILLIAM	15.00
	7000227	S FE EMP HOSP ASSN	36.00
	6211151	COX,WILLIAM	100.20
	6208924	WHITE,ELMER	15.00
	6215785	RODEN,HAROLD	170.91
	9002626	WORRELL, TED	13.00

Figure 18 Report with Account Codes Suppressed Between Changes (Page 2 of 2)

Printing Total at Control Break

In this example, a control break occurs whenever there is a change in the value of ACCOUNT.

```
FILE ARFILE FB 352 5280
    ACCOUNT 2 182   NAME 25 85   TRANS 7 4
    BALANCE 5 170 PD 2 A
CONTROL ACCOUNT
SORT ARFILE USING ACCOUNT
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE
ON CHANGE IN ACCOUNT
   LIST SUM BALANCE WITH 2 BEFORE AND 2 AFTER
T1 'VISION:RESULTS REPORT 3'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 19 Print Total at Control Break

The total of BALANCE (indicated by the keyword SUM) is printed at each control break, with double spacing occurring both before and after the control break print line.

```
12/01/00                VISION:RESULTS REPORT 3              PAGE 1

 ACCOUNT      TRANS              NAME                      BALANCE

    BO       8006547    TORRES,ERNESTO                        44.99

                                                              44.99

    EO       6208657    CHO PYUNG,SUH                         32.00
             6002587    SANTA FE HOSP ASSN                    15.00
             7082509    S.F.MEM.HOSP.                          5.00

                                                              52.00

    FO       8011508    HUGHES,RAY                           178.70
             6024963    HILL,GARY E                            3.80
             6044395    CANO,MICHAEL S                        15.00
             6059708    CHAVEZ,RAY                            15.00
             6123228    SILVA,JULIAN                           .00
             6095631    TODIPE,MICHAEL                        45.24
             6107265    GENVARDI,G J                          43.00

                                                             300.74

    IO       2002299    PLACIDO,ORTEGA                       413.58
             6218113    AGUIERA,EMILIO                       108.44
             8012644    MONTEZ,CARMEN                         89.28



12/01/00                VISION:RESULTS REPORT 3              PAGE 6

 ACCOUNT      TRANS              NAME                      BALANCE

    WO       8011699    OBRESON,FRANK                         57.50
             7090943    KANGAROO R.ROAD CO.                   11.20
             7071361    COMPLIANCE INSURANCE                  66.00
             7112947    R L KAUTZ INS                          .00
             7098375    KANGAROO R.ROAD CO.                   22.40
             9009469    EMP MUT OF WAUSAU                     72.50
             9007156    KANGAROO R.ROAD CO.                  352.80
             8031096    ROMERO,ARMENDO                        81.26
             8010641    MURPHY,MATTHEWS                       82.56
             6118577    NAULLS,MARK A                         15.00
             6206328    VILLASENOR,TERESA                     15.00
             6019587    BENAVIDEZ,CESARIO                     15.00
             6032214    GONZALEZ,RAUL                         30.40
             6044832    ROBBINS,WILLIAM D                      .00
             6066402    ALCARAZ,ANDRES                        32.80
             6074359    ASHLEY,JOHN                           15.00
             7043732    INS CO OF NO AMER                    108.25
             7054645    KANGAROO R.ROAD CO.                   22.40
             7059582                                          44.75
             9020039    LIBERTY MUTUAL INS                    58.24
             7093853    KANGAROO R.ROAD CO.                  265.74
             7084447    KANGAROO R.ROAD CO.                  121.55
             7097166    LIBERTY MUTUAL INS                    44.45

                                                           2,034.15
```

Figure 20 Report with Totals at Control Breaks

Copying a File Definition

In this example, the FILE statement and field definitions are replaced with a COPY statement followed by the member name. This member name contains not only the FILE statement and field definitions shown in the previous three examples, but the definitions of all the fields in the record as well.

```
COPY ARDEF
CONTROL ACCOUNT
 SORT ARFILE USING ACCOUNT
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE
ON CHANGE IN ACCOUNT
   LIST SUM BALANCE WITH 2 BEFORE AND 2 AFTER
T1 'VISION:RESULTS REPORT 4'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 21 Copy File Definition Example 1

Note: Use of member name ARDEF is for example purposes only. It is not supplied with your VISION:Results system.

If the field definitions are already defined as COBOL data definitions, you could code the following (assuming that ARDEF contains COBOL data definitions) and use these existing definitions.

```
FILE ARFILE FB 352 5280
COPY ARDEF COBOL
CONTROL ACCOUNT
SORT ARFILE USING ACCOUNT
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE
ON CHANGE IN ACCOUNT
   LIST SUM BALANCE WITH 2 BEFORE AND 2 AFTER
T1 'VISION:RESULTS REPORT 4'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 22 Copy File Definition Example 2

```
12/01/00              VISION:RESULTS REPORT 4              PAGE 1

ACCOUNT      TRANS          NAME                        BALANCE

  BO       8006547    TORRES,ERNESTO                        44.99

                                                           44.99

  EO       6208657    CHO PYUNG,SUH                         32.00
           6002587    SANTA FE HOSP ASSN                    15.00
           7082509    S.F.MEM.HOSP.                          5.00

                                                           52.00

  FO       8011508    HUGHES,RAY                           178.70
           6024963    HILL,GARY E                            3.80
           6044395    CANO,MICHAEL S                        15.00
           6059708    CHAVEZ,RAY                            15.00
           6123228    SILVA,JULIAN                            .00
           6095631    TODIPE,MICHAEL                        45.24
           6107265    GENVARDI,G J                          43.00

                                                          300.74

  IO       2002299    PLACIDO,ORTEGA                       413.58
           6218113    AGUIERA,EMILIO                       108.44
           8012644    MONTEZ,CARMEN                         89.28
           6009166    LOCKE,JEFFREY                         15.00
           6112536    CHAVEZ,NORMA A                        55.00
           6123317    VASGUEZ,IRENE                           .00
12/01/00              VISION:Results REPORT 4              PAGE 6

ACCOUNT      TRANS          NAME                        BALANCE

  WO       8011699    OBRESON,FRANK                         57.50
           7090943    KANGAROO R.ROAD CO.                   11.20
           7071361    COMPLIANCE INSURANCE                  66.00
           7112947    R L KAUTZ INS                           .00
           7098375    KANGAROO R.ROAD CO.                   22.40
           9009469    EMP MUT OF WAUSAU                     72.50
           9007156    KANGAROO R.ROAD CO.                  352.80
           8031096    ROMERO,ARMENDO                        81.26
           8010641    MURPHY,MATTHEWS                       82.56
           6118577    NAULLS,MARK A                         15.00
           6206328    VILLASENOR,TERESA                     15.00
           6019587    BENAVIDEZ,CESARIO                     15.00
           6032214    GONZALEZ,RAUL                         30.40
           6044832    ROBBINS,WILLIAM D                       .00
           6066402    ALCARAZ,ANDRES                        32.80
           6074359    ASHLEY,JOHN                           15.00
           7043732    INS CO OF NO AMER                    108.25
           7054645    KANGAROO R.ROAD CO.                   22.40
           7059582                                          44.75
           9020039    LIBERTY MUTUAL INS                    58.24
           7093853    KANGAROO R.ROAD CO.                  265.74
           7084447    KANGAROO R.ROAD CO.                  121.55
           7097166    LIBERTY MUTUAL INS                    44.45

                                                        2,034.15
```

Figure 23 Report Using the COPY Statement

Printing Total and Tally

The example below differs in two ways from the previous example. First, the total of BALANCE for all accounts is printed on the report when end of processing occurs. Second, the number of records printed is listed at the end of the report using the reserved word counter TALLY. The FILE statement and field definitions are again used for this and the following examples.

```
FILE ARFILE FB 352 5280
    ACCOUNT 2 182   NAME 25 85   TRANS 7 4
    BALANCE 5 170 PD 2 A
CONTROL ACCOUNT
SORT ARFILE USING ACCOUNT
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE
ON CHANGE IN ACCOUNT
   LIST SUM BALANCE WITH 2 BEFORE AND 2 AFTER
ON FINAL
   LIST SUM BALANCE WITH 2 AFTER
   LIST 'NUMBER OF RECORDS PRINTED:' AT ACCOUNT
   TALLY AT BALANCE
T1 'VISION:RESULTS REPORT 5'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 24 Print Total and Tally

```
12/01/00              VISION:RESULTS REPORT 5              PAGE 1

 ACCOUNT     TRANS             NAME                  BALANCE

    BO       8006547    TORRES,ERNESTO                   44.99

                                                         44.99

    EO       6208657    CHO PYUNG,SUH                    32.00
             6002587    SANTA FE HOSP ASSN               15.00
             7082509    S.F.MEM.HOSP.                     5.00

                                                         52.00

    FO       8011508    HUGHES,RAY                      178.70
             6024963    HILL,GARY E                       3.80
             6044395    CANO,MICHAEL S                   15.00
             6059708    CHAVEZ,RAY                       15.00
             6123228    SILVA,JULIAN                      .00
             6095631    TODIPE,MICHAEL                   45.24
             6107265    GENVARDI,G J                     43.00

                                                        300.74

    IO       2002299    PLACIDO,ORTEGA                  413.58
             6218113    AGUIERA,EMILIO                  108.44
```

Figure 25 Report Showing Balance Total and Records Tallied (Page 1 of 2)

```
12/01/00              VISION:RESULTS REPORT 5              PAGE 6

 ACCOUNT      TRANS              NAME                  BALANCE

   WO        8011699    OBRESON,FRANK                     57.50
             7090943    KANGAROO R.ROAD CO.               11.20
             7071361    COMPLIANCE INSURANCE              66.00
             7112947    R L KAUTZ INS                       .00
             7098375    KANGAROO R.ROAD CO.               22.40
             9009469    EMP MUT OF WAUSAU                 72.50
             9007156    KANGAROO R.ROAD CO.              352.80
             8031096    ROMERO,ARMENDO                    81.26
             8010641    MURPHY,MATTHEWS                   82.56
             6118577    NAULLS,MARK A                     15.00
             6206328    VILLASENOR,TERESA                 15.00
             6019587    BENAVIDEZ,CESARIO                 15.00
             6032214    GONZALEZ,RAUL                     30.40
             6044832    ROBBINS,WILLIAM D                   .00
             6066402    ALCARAZ,ANDRES                    32.80
             6074359    ASHLEY,JOHN                       15.00
             7043732    INS CO OF NO AMER                108.25
             7054645    KANGAROO R.ROAD CO.               22.40
             7059582                                      44.75
             9020039    LIBERTY MUTUAL INS                58.24
             7093853    KANGAROO R.ROAD CO.              265.74
             7084447    KANGAROO R.ROAD CO.              121.55
             7097166    LIBERTY MUTUAL INS                44.45

                                                        2,034.15

                                                       19,023.10

 NUMBER OF RECORDS PRINTED:                                200
```

Figure 25 Report Showing Balance Total and Records Tallied (Page 2 of 2)

Selecting Specific Records

In the previous examples, the entire file was sorted on account code and then reported on. In this example, the program selects, sorts, and reports on only those records with account codes equal to NA. The sort sequence is TRANS (minor) to ACCOUNT (major).

```
FILE ARFILE FB 352 5280
    ACCOUNT 2 182   NAME 25 85   TRANS 7 4
    BALANCE 5 170 PD 2 A
CONTROL ACCOUNT
IF ACCOUNT  EQ  'NA'
NEXT
   ELSE REJECT ENDIF
SORT ARFILE USING ACCOUNT TRANS
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE
 ON CHANGE IN ACCOUNT
   LIST SUM BALANCE WITH 2 BEFORE AND 2 AFTER
ON FINAL
   LIST SUM BALANCE  WITH 2  AFTER
   LIST 'NUMBER OF RECORDS PRINTED:' AT ACCOUNT
   TALLY AT BALANCE
T1 'VISION:RESULTS REPORT 6'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 26 Select Specific Records

```
12/01/00              VISION:RESULTS REPORT 6              PAGE 1

 ACCOUNT     TRANS              NAME                 BALANCE

   NA       0000434                                     11.20-
            0002623                                    256.12-
            1009681                                 10,085.55-
            6027989                                    310.75
            6045154                                     64.96
            6090516                                     19.54
            7046936                                      2.46-
            7053941                                     33.60-
            7066775                                      3.32-
            7087896                                      1.32-
            8003173                                     35.00
            9010033                                      1.04-

                                                     9,964.36-
                                                     9,964.36-

 NUMBER OF RECORDS PRINTED:                             12
```

Figure 27 Report of Selected Account Codes

Selecting Multiple Records

As with the preceding examples, the program selects and sorts specific records. The program selects account codes of NA and EO through IO, inclusive (EO, EP through EZ, FA, FB through FZ, and up to and including IO).

```
FILE ARFILE FB 352 5280
   ACCOUNT 2 182   NAME 25 85   TRANS 7 4
   BALANCE 5 170 PD 2 A
CONTROL ACCOUNT
IF ACCOUNT EQ 'NA'    'EO' THRU 'IO'  NEXT
   ELSE REJECT
ENDIF
SORT ARFILE USING ACCOUNT TRANS
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE
ON CHANGE IN ACCOUNT
   LIST SUM BALANCE WITH 2 BEFORE AND 2 AFTER
ON FINAL
    LIST SUM BALANCE  WITH 2  AFTER
   LIST 'NUMBER OF RECORDS PRINTED:' AT ACCOUNT
   TALLY AT BALANCE
T1 'VISION:RESULTS REPORT 7'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 28 Select Multiple Records

```
12/01/00              VISION:RESULTS REPORT 7              PAGE 1

 ACCOUNT     TRANS              NAME                 BALANCE

   EO       6002587    SANTA FE HOSP ASSN                15.00
            6208657    CHO PYUNG,SUH                     32.00
            7082509    S.F.MEM.HOSP.                      5.00

                                                         52.00
```

Figure 29 Report of Multiple Selected Records (Page 1 of 2)

```
   FO       6024963    HILL,GARY E                      3.80
            6044395    CANO,MICHAEL S                  15.00
            6059708    CHAVEZ,RAY                      15.00
            6095631    TODIPE,MICHAEL                  45.24
            6107265    GENVARDI,G J                    43.00
            6123228    SILVA,JULIAN                      .00
            8011508    HUGHES,RAY                     178.70

                                                      300.74

   IO       2002299    PLACIDO,ORTEGA                 413.58
            6009166    LOCKE,JEFFREY                   15.00
            6112536    CHAVEZ,NORMA A                  55.00
            6123317    VASQUEZ,IRENE                     .00
            6218113    AGUIERA,EMILIO                 108.44
            8012644    MONTEZ,CARMEN                   89.28

                                                      681.30

   NA       0000434                                    11.20-
            0002623                                   256.12-
            1009981                                10,085.55-
            6027989                                   310.75
            6045154                                    64.96
            6090516                                    19.54
            7046936                                     2.46-
            7053941                                    33.60-
            7066775                                     3.32-
            7087896                                     1.32-
            8003173                                    35.00
            9010033                                     1.04-

                                                    9,964.36-

                                                    8,930.32-

  NUMBER OF RECORDS PRINTED:                           28
```

Figure 29 Report of Multiple Selected Records (Page 2 of 2)

Using Conditional Expression and WORKAREA

In this example, the program prints only those records with account codes NA or EO through IO inclusive. All records in this group with a balance greater than or equal to $300.00 are flagged with *** in the column labeled BALANCE OVER $300.

This example introduces the WORKAREA. Here you can define area definitions, such as literals and counters, for your own use. You can give them column headings, initialize the field definition to a value (the above bytes were initialized to spaces), or cause VISION:Results to reinitialize that value after each cycle to its original value if you use the keyword REINIT.

```
FILE ARFILE FB 352 5280
    ACCOUNT 2 182   NAME 25 85   TRANS 7 4
    BALANCE 5 170 PD 2 A
WORKAREA  COMMENT 3 (BALANCE'OVER $300.00)
    VALUE  ' '    REINIT
CONTROL ACCOUNT
IF ACCOUNT EQ 'NA'   'EO' THRU 'IO'   NEXT
   ELSE REJECT
ENDIF
   SORT ARFILE USING ACCOUNT TRANS
IF BALANCE  GE  300
   MOVE '***' TO COMMENT
ENDIF
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE COMMENT
```

Figure 30 Conditional Expression and WORKAREA (Page 1 of 2)

```
ON CHANGE IN ACCOUNT
   LIST SUM BALANCE WITH 2 BEFORE AND 2 AFTER
ON FINAL
   LIST SUM BALANCE  WITH 2  AFTER
   LIST 'NUMBER OF RECORDS PRINTED:' AT ACCOUNT
   TALLY AT BALANCE
T1 'VISION:RESULTS REPORT 8'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 30 Conditional Expression and WORKAREA (Page 2 of 2)

```
    12/01/00              VISION:RESULTS REPORT 8            PAGE          1

ACCOUNT      TRANS              NAME                  BALANCE         BALANCE
                                                                    OVER $300.00

  EO       6002587    SANTA FE HOSP ASSN                 15.00
           6208657    CHO PYUNG,SUH                      32.00
           7082509    S.F.MEM.HOSP.                       5.00

                                                         52.00

  FO       6024963    HILL,GARY E                         3.80
           6044395    CANO,MICHAEL S                     15.00
           6059708    CHAVEZ,RAY                         15.00
           6095631    TODIPE,MICHAEL                     45.24
           6107265    GENVARDI,G J                       43.00
           6123228    SILVA,JULIAN                         .00
           8011508    HUGHES,RAY                        178.70

                                                        300.74

  IO       2002299    PLACIDO,ORTEGA                    413.58        ***
           6009166    LOCKE,JEFFREY                      15.00
           6112536    CHAVEZ,NORMA A                     55.00
           6123317    VASQUEZ,IRENE                        .00
           6218113    AGUIERA,EMILIO                    108.44
           8012644    MONTEZ,CARMEN                      89.28

                                                        681.30

  NA       0000434                                       11.20-
           0002623                                      256.12-
           1009681                                   10,085.55-
           6027989                                      310.75        ***
           6045154                                       64.96
           6090516                                       19.54
           7046936                                        2.46-
           7053941                                       33.60-
           7066775                                        3.32-
           7087896                                        1.32-
           8003173                                       35.00
           9010033                                        1.04-

                                                     9,964.36-

                                                     8,930.32-

NUMBER OF RECORDS PRINTED:                              28
```

Figure 31 Report Using Conditional Expression and WORKAREA

Using Multiple Conditional Expressions

In this example, the program also flags those records whose balance is less than zero. The comment column heading has been changed to include these accounts, flagged with ---. The GOTO, as well as the tag name reference point capabilities of VISION:Results (LISTDET:), has been introduced.

```
FILE ARFILE FB 352 5280
    ACCOUNT 2 182   NAME 25 85   TRANS 7 4
    BALANCE 5 170 PD 2 A
WORKAREA COMMENT 3 (BALANCE'GE $300.00'OR'LT $0.00)
         VALUE ' '   REINIT
CONTROL ACCOUNT
IF ACCOUNT EQ 'NA'   'EO' THRU 'IO'   NEXT
   ELSE REJECT
ENDIF
     SORT ARFILE USING ACCOUNT TRANS
IF BALANCE   GE    300
   MOVE '***' TO COMMENT
   GO TO LISTDET
ENDIF
IF BALANCE   LT   0
   MOVE '--' TO COMMENT
ENDIF
LISTDET:
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE COMMENT
ON CHANGE IN ACCOUNT
   LIST SUM BALANCE WITH 2 BEFORE AND 2 AFTER
ON FINAL
   LIST SUM BALANCE  WITH 2  AFTER
   LIST 'NUMBER OF RECORDS PRINTED:' AT ACCOUNT
   TALLY AT BALANCE
T1 'VISION:RESULTS REPORT 9'
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 32 Multiple Conditional Expressions

```
12/01/00              VISION:RESULTS REPORT 9              PAGE       1

ACCOUNT     TRANS              NAME                BALANCE        BALANCE
                                                                GE $300.00
                                                                    OR
                                                                LT $0.00

  EO       6002587     SANTA FE HOSP ASSN              15.00
           6208657     CHO PYUNG,SUH                   32.00
           7082509     S.F.MEM.HOSP.                    5.00

                                                       52.00

  FO       6024963     HILL,GARY E                      3.80
           6044395     CANO,MICHAEL S                  15.00
           6059708     CHAVEZ,RAY                      15.00
           6095631     TODIPE,MICHAEL                  45.24
           6107265     GENVARDI,G J                    43.00
           6123228     SILVA,JULIAN                      .00
           8011508     HUGHES,RAY                     178.70

                                                      300.74

  IO       2002299     PLACIDO,ORTEGA                 413.58       ***
           6009166     LOCKE,JEFFREY                   15.00
           6112536     CHAVEZ,NORMA A                  55.00
           6123317     VASGUEZ,IRENE                     .00
           6218113     AGUIERA,EMILIO                 108.44
           8012644     MONTEZ,CARMEN                   89.28

                                                      681.30
```

Figure 33 Report Using Multiple Conditional Expressions (Page 1 of 2)

```
 NA       0000434                                  11.20-      —
          0002623                                 256.12-      —
          1009681                              10,085.55-      —
          6027989                                 310.75       ***
          6045154                                  64.96
          6090516                                  19.54
          7046936                                   2.46-      —
          7053941                                  33.60-      —
          7066775                                   3.32-      —
          7087896                                   1.32-      —
          8003173                                  35.00
          9010033                                   1.04-      —

                                                9,964.36-

                                                8,930.32-

NUMBER OF RECORDS PRINTED:                         28
```

Figure 33 Report Using Multiple Conditional Expressions (Page 2 of 2)

Assigning an Arithmetic Expression to a Variable

In this example, a surcharge of 10% has been added to those records with a balance greater than or equal to $300.00. Both the original BALANCE and the BALANCE WITH SURCHARGE are printed on the report. The actual input file is left unchanged.

```
FILE ARFILE FB 352 5280
    ACCOUNT 2 182   NAME 25 85   TRANS 7 4
    BALANCE 5 170 PD 2 A
WORKAREA COMMENT 3 (BALANCE'GE $300.00'OR'LT $0.00)
         VALUE ' '   REINIT
    JBALANCE 5   PD 2 A (BALANCE WITH'SURCHARGE)
         VALUE 0 REINIT
CONTROL ACCOUNT
IF ACCOUNT EQ 'NA'   'EO' THRU 'IO'   NEXT
   ELSE REJECT
ENDIF
   SORT ARFILE USING ACCOUNT TRANS
IF BALANCE   GE   300
   MOVE '***' TO COMMENT
   JBALANCE = BALANCE *1.10
   GOTO LISTDET
ENDIF
IF BALANCE   LT   0
   MOVE '--' TO COMMENT
ENDIF
LISTDET:
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE JBALANCE COMMENT
ON CHANGE IN ACCOUNT
   LIST SUM BALANCE WITH 2 BEFORE AND 2 AFTER
ON FINAL
    LIST SUM BALANCE  WITH 2  AFTER
   LIST 'NUMBER OF RECORDS PRINTED:' AT ACCOUNT
   TALLY AT BALANCE
T1 'VISION:RESULTS REPORT 10'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 34 Assigning an Arithmetic Expression to a Variable

```
12/01/00              VISION:RESULTS REPORT 10              PAGE       1

ACCOUNT     TRANS          NAME            BALANCE      BALANCE WITH    BALANCE
                                                         SURCHARGE       GE $300.00
                                                                            OR
                                                                         LT $0.00

  EO        6002587    SANTA FE HOSP ASSN      15.00          .00
            6208657    CHO PYUNG,SUH           32.00          .00
            7082509    S.F.MEM.HOSP.            5.00          .00

                                               52.00

  FO        6024963    HILL,GARY E              3.80          .00
            6044395    CANO,MICHAEL S          15.00          .00
            6059708    CHAVEZ,RAY              15.00          .00
            6095631    TODIPE,MICHAEL          45.24          .00
            6107265    GENVARDI,G J            43.00          .00
            6123228    SILVA,JULIAN              .00          .00
            8011508    HUGHES,RAY             178.70          .00

                                              300.74

  IO        2002299    PLACIDO,ORTEGA         413.58       454.93       ***
            6009166    LOCKE,JEFFREY           15.00          .00
            6112536    CHAVEZ,NORMA A          55.00          .00
            6123317    VASQUEZ,IRENE             .00          .00
            6218113    AGUIERA,EMILIO         108.44          .00
            8012644    MONTEZ,CARMEN           89.28          .00

                                              681.30

  NA        0000434                         11.20-          .00       —
            0002623                        256.12-          .00       —
            1009681                     10,085.55-          .00       —
            6027989                        310.75       341.82       ***
            6045154                         64.96          .00
            6090516                         19.54          .00
            7046936                          2.46-         .00       —
            7053941                         33.60-         .00       —
            7066775                          3.32-         .00       —
            7087896                          1.32-         .00       —
            8003173                         35.00          .00
            9010033                          1.04-         .00       —

                                            9,964.36-

                                            8,930.32-

NUMBER OF RECORDS PRINTED:                      28
```

Figure 35 Report with an Arithmetic Expression Assigned to a Variable

Flagging an Updated Field

In this example, an output file is created with the BALANCE field updated to include a surcharge of 10% on those records whose balance is greater than or equal to $300.00.

```
FILE ARFILE FB 352 5280
    ACCOUNT 2 182   NAME 25 85   TRANS 7 4
    BALANCE 5 170 PD 2 A
FILE OARFILE FB 352 5280 OUTPUT FROM ARFILE
WORKAREA COMMENT 3 (BALANCE'GE $300.00'OR'LT'$0.00)
         VALUE ' '   REINIT
CONTROL ACCOUNT
IF ACCOUNT  EQ  'NA'  'EO'  THRU  'IO'  NEXT
   ELSE GOTO  WRT
ENDIF
IF BALANCE   GE   300
   BALANCE = BALANCE * 1.10
ENDIF

WRT:    WRITE  OARFILE

IF ACCOUNT  EQ  'NA'  'EO'  THRU 'IO'   NEXT
   ELSE  REJECT
ENDIF
SORT ARFILE  USING  ACCOUNT  TRANS
IF BALANCE   GE   300
   MOVE '***' TO COMMENT
   GOTO  LISTDET
ENDIF
IF BALANCE LT ZERO
   MOVE '--' TO COMMENT
ENDIF
LISTDET:
LIST SUPPRESS ACCOUNT TRANS NAME BALANCE COMMENT
ON CHANGE IN ACCOUNT
   LIST SUM BALANCE WITH 2 BEFORE AND 2 AFTER
ON FINAL
   LIST SUM BALANCE  WITH 2  AFTER
   LIST 'NUMBER OF RECORDS PRINTED:' AT ACCOUNT
   TALLY AT BALANCE
T1 'VISION:RESULTS REPORT 11'  WITH 2 AFTER
T1+30 DYLDATE
T1+90 DYLPAGE
```

Figure 36 Flagging an Updated Field

Note: Although the report contains only those records with an account code of NA or EO through IO inclusive, the output file contains all records, updated or not, on the original file.

On the report, the updated fields have been flagged.

```
12/01/00              VISION:RESULTS REPORT 11              PAGE       1

ACCOUNT     TRANS              NAME                 BALANCE        BALANCE
                                                                  GE $300.00
                                                                     OR
                                                                  LT $0.00


  EO       6002587    SANTA FE HOSP ASSN                15.00
           6208657    CHO PYUNG,SUH                     32.00
           7082509    S.F.MEM.HOSP.                      5.00

                                                        52.00
```

Figure 37 Report with Flagged Updated Fields (Page 1 of 2)

```
 FO       6024963    HILL,GARY E                      3.80
          6044395    CANO,MICHAEL S                  15.00
          6059708    CHAVEZ,RAY                      15.00
          6095631    TODIPE,MICHAEL                  45.24
          6107265    GENVARDI,G J                    43.00
          6123228    SILVA,JULIAN                      .00
          8011508    HUGHES,RAY                     178.70

                                                     300.74

 IO       2002299    PLACIDO,ORTEGA                 413.58      ***
          6009166    LOCKE,JEFFREY                   15.00
          6112536    CHAVEZ,NORMA A                  55.00
          6123317    VASGUEZ,IRENE                     .00
          6218113    AGUIERA,EMILIO                 108.44
          8012644    MONTEZ,CARMEN                   89.28

                                                     681.30

 NA       0000434                                    11.20-     —
          0002623                                   256.12-     —
          1009681                                10,085.55-     —
          6027989                                   310.75      ***
          6045154                                    64.96
          6090516                                    19.54
          7046936                                     2.46-     —
          7053941                                    33.60-     —
          7066775                                     3.32-     —
          7087896                                     1.32-     —
          8003173                                    35.00
          9010033                                     1.04-     —

                                                   9,964.36-

                                                   8,930.32-

NUMBER OF RECORDS PRINTED:                            28
```

Figure 37 Report with Flagged Updated Fields (Page 2 of 2)

Using Summary Recap Reporting with Variable Data

This example demonstrates summary recap reporting. An optional date is passed to the reserved word DYLPARM using the keyword DATA in the OPTION statement.

Note: DYLPARM is a 60-byte area in VISION:Results. The DATA keyword in the OPTION statement can pass up to 38 bytes of information to a program. z/OS users have the added option of the z/OS PARM facility in the EXEC statement.

```
OPTION  DATA  '12/01/00'
FILE ARFILE  FB  352  5280
     ACCOUNT  2  182   BALANCE 5  170  PD 2 A
     INSTLBAL 6  191 PD 2 A (INSTALL'BALANCE)
     INSTPAY  5  197 PD 2 A (INSTALL'PAY)
     BALPART  4  202 PD 2 A (BALANCE'PART'PAY)
     INTPART  3  206 PD 2 A (INTEREST'PART'PAY)
WORKAREA  TODAYSDATE  8
ON ONE
      MOVE  DYLPARM  TO  TODAYSDATE
ENDONE
      READ ARFILE
      SORT  ARFILE  USING  ACCOUNT
ON FINAL
      LIST SUM BALANCE SUM INSTLBAL SUM INSTPAY
           SUM BALPART SUM INTPART
T1 'VISION:RESULTS REPORT 12'  WITH  2  AFTER
T1+30 TODAYSDATE
T1+90 DYLPAGE
```

Figure 38 Recap Report with Variable Data

To access the data in DYLPARM, move it (only 8 bytes are needed) to a work area definition (in this example, TODAYSDATE). This happens only once during processing because MOVE DYLPARM ... is between the ON ONE and ENDONE commands.

This example also introduces the immediate READ command.

```
12/01/00              VISION:RESULTS REPORT 12              PAGE       1

BALANCE            INSTALL            INSTALL          BALANCE       INTEREST
                   BALANCE              PAY              PART          PART
                                                         PAY           PAY

  19,023.10         145,346.73         10,451.64        9,841.02      610.62
```

Figure 39 Report Using Summary Recap Reporting and the READ Command

Using Recap Reporting with Account Summaries

In this example, the printing of a control break line has been added for any change in the field ACCOUNT. A summary recap by account code (ACCOUNT) and a grand total for all accounts are printed.

```
OPTION  DATA  '12/01/00'
FILE ARFILE  FB  352  5280
      ACCOUNT  2  182   BALANCE 5  170  PD 2 A
     INSTLBAL 6  191 PD 2 A (INSTALL'BALANCE)
     INSTPAY  5  197 PD 2 A (INSTALL'PAY)
     BALPART  4  202 PD 2 A (BALANCE'PART'PAY)
     INTPART  3  206 PD 2 A (INTEREST'PART'PAY)
WORKAREA  TODAYSDATE  8
ON ONE
     MOVE  DYLPARM  TO  TODAYSDATE
ENDONE
     READ  ARFILE
     SORT  ARFILE  USING  ACCOUNT
CONTROL  ACCOUNT
ON CHANGE IN ACCOUNT
     LIST ACCOUNT SUM BALANCE SUM INSTLBAL SUM INSTPAY
       SUM BALPART SUM INTPART  WITH 2 BEFORE AND 2 AFTER
ON FINAL
     LIST SUM BALANCE SUM INSTLBAL SUM INSTPAY
       SUM BALPART SUM INTPART  WITH 1 BEFORE
T1 'VISION:RESULTS REPORT 13'  WITH 2 AFTER
T1+30 TODAYSDATE
T1+90 DYLPAGE
```

Figure 40 Recap Reporting with Account Summaries

12/01/00 VISION:RESULTS REPORT 13 PAGE 1

ACCOUNT	BALANCE	INSTALL BALANCE	INSTALL PAY	BALANCE PART PAY	INTEREST PART PAY
BO	44.99	42.41	.00	.00	.00
EO	52.00	2,764.52	193.60	181.98	11.62
FO	300.74	1,495.07	234.76	229.24	5.52
IO	681.30	2,396.96	409.12	399.50	9.62
KO	3,765.92	19,441.70	1,223.76	1,142.12	81.64
LO	45.24	298.66	50.98	49.78	1.20
MA	8,524.80	43,788.39	2,929.69	2,745.46	184.23
MF	1,836.67	9,572.50	541.29	500.36	40.93
MI	276.23	538.18	79.23	77.37	1.86

12/01/00 VISION:RESULTS REPORT 13 PAGE 2

ACCOUNT	BALANCE	INSTALL BALANCE	INSTALL PAY	BALANCE PART PAY	INTEREST PART PAY
MT	359.13	1,580.25	269.71	263.38	6.33
MU	629.05	718.97	113.38	110.72	2.66
NA	9,964.36-	16,047.91	861.69	792.60	69.09
PO	440.10	5,596.61	759.44	737.38	22.06
RO	65.00	203.55	34.74	33.93	.81
TO	6,875.90	7,559.31	460.51	428.03	32.48
VO	41.91	100.98	17.24	16.83	.41
WE	11.20	268.64	45.85	44.77	1.08
WO	2,034.15	17,526.50	1,499.54	1,426.59	72.95
	19,023.10	145,346.73	10,451.64	9,841.02	610.62

Figure 41 Recap Report with Account Summaries

Using Recap Reporting with Calculation Averages

This example introduces processing at control break time (ON CHANGE IN). Whenever there is a change in the value of ACCOUNT, VISION:Results calculates five separate averages, using the SUM of all dollar fields involved and the TALLY counter provided by VISION:Results. In addition, a second line containing these calculated averages prints at control break time.

```
OPTION  DATA  '12/01/00'
FILE ARFILE  FB  352  5280
     ACCOUNT  2  182   BALANCE 5  170  PD 2 A
     INSTLBAL 6  191 PD 2 A (INSTALL'BALANCE)
     INSTPAY  5  197 PD 2 A (INSTALL'PAY)
     BALPART  4  202 PD 2 A (BALANCE'PART'PAY)
     INTPART  3  206 PD 2 A (INTEREST'PART'PAY)
WORKAREA
     AVG1    5  PD  2  E   VALUE 0
     AVG2    5  PD  2  E   VALUE 0
     AVG3    5  PD  2  E   VALUE 0
     AVG4    5  PD  2  E   VALUE 0
     AVG5    4  PD  2  E   VALUE 0
WORKAREA  TODAYSDATE  8
ON ONE
      MOVE  DYLPARM  TO  TODAYSDATE
ENDONE
      READ  ARFILE
      SORT  ARFILE  USING  ACCOUNT
CONTROL ACCOUNT
ON CHANGE  IN ACCOUNT
* CALCULATE AN AVERAGE FOR EACH DOLLAR FIELD THAT IS TOTALED
  AVG1=SUM BALANCE/TALLY   ;CALCULATE BALANCE AVERAGE
  AVG2=SUM INSTLBAL/TALLY  ;CALCULATE INSTALLMENT BALANCE AVERAGE
  AVG3=SUM INSTPAY/TALLY   ;CALCULATE INSTALLMENT PARTIAL AVERAGE
  AVG4=SUM BALPART/TALLY   ;CALCULATE BALANCE PARTIAL AVERAGE
  AVG5=SUM INTPART/TALLY   ;CALCULATE INTEREST PARTIAL AVERAGE
  LIST ACCOUNT SUM BALANCE SUM INSTLBAL SUM INSTPAY SUM BALPART
       SUM INTPART   WITH 2 BEFORE AND 2 AFTER
 LIST 'AVERAGE' AT ACCOUNT AVG1 AVG2 AVG3 AVG4 AVG5 WITH 2 AFTER
ON FINAL
  LIST  SUM BALANCE SUM INSTLBAL SUM INSTPAY SUM BALPART
       SUM INTPART   WITH 2 BEFORE
T1 'VISION:RESULTS REPORT 14'  WITH 2 AFTER
 T1+30 TODAYSDATE
T1+90 DYLPAGE
```

Figure 42 Recap by Account with Calculation Averages

This example also demonstrates the use of the asterisk (*) and semicolon (;) to indicate comments.

12/01/00 VISION:RESULTS REPORT 14 PAGE 1

ACCOUNT	BALANCE	INSTALL BALANCE	INSTALL PAY	BALANCE PART PAY	INTEREST PART PAY
BO	44.99	42.41	.00	.00	.00
AVERAGE	44.99	42.41			
EO	52.00	2,764.52	193.60	181.98	11.62
AVERAGE	17.33	921.50	64.53	60.66	3.87
FO	300.74	1,495.07	234.76	229.24	5.52
AVERAGE	42.96	213.58	33.53	32.74	.78
IO	681.30	2,396.96	409.12	399.50	9.62
AVERAGE	113.55	399.49	68.18	66.58	1.60
KO	3,765.92	19,441.70	1,223.76	1,142.12	81.64

12/01/00 VISION:RESULTS REPORT 14 PAGE 3

ACCOUNT	BALANCE	INSTALL BALANCE	INSTALL PAY	BALANCE PART PAY	INTEREST PART PAY
TO	6,875.90	7,559.31	460.51	428.03	32.48
AVERAGE	982.27	1,079.90	65.78	61.14	4.64
VO	41.91	100.98	17.24	16.83	.41
AVERAGE	41.91	100.98	17.24	16.83	.41
WE	11.20	268.64	45.85	44.77	1.08
AVERAGE	11.20	268.64	45.85	44.77	1.08
WO	2,034.15	17,526.50	1,499.54	1,426.59	72.95
AVERAGE	72.64	625.94	53.55	50.94	2.60
	19,023.10	145,346.73	10,451.64	9,841.02	610.62

Figure 43 Recap Report by Account with Calculation Averages

Creating Summary Output File and Recap by Account

This example further exemplifies the flexibility of VISION:Results. Here, the keyword ROUNDED is added to each average being calculated.

```
OPTION  DATA  '12/01/00'
FILE ARFILE  FB  352  5280
     ACCOUNT  2  182   BALANCE 5  170  PD 2 A
     INSTLBAL 6  191 PD 2 A (INSTALL'BALANCE)
     INSTPAY  5  197 PD 2 A (INSTALL'PAY)
     BALPART  4  202 PD 2 A (BALANCE'PART'PAY)
     INTPART  3  206 PD 2 A (INTEREST'PART'PAY)
FILE OUTFILE   FB  25   2500  OUTPUT FROM OUTFILE
      OUTACCT   2   OUTBALANCE  5  PD  2   OUTINSTAL  6  PD  2
      OUTINSTPAY 5 PD  2 OUTBALPART 4 PD 2  OUTINTPART 3  PD 2

WORKAREA
     AVG1   5  PD  2  E   VALUE 0
     AVG2   5  PD  2  E   VALUE 0
     AVG3   5  PD  2  E   VALUE 0
     AVG4   5  PD  2  E   VALUE 0
     AVG5   4  PD  2  E   VALUE 0
WORKAREA  TODAYSDATE  8
ON ONE
      MOVE  DYLPARM  TO  TODAYSDATE
ENDONE
      READ  ARFILE
      SORT  ARFILE  USING  ACCOUNT
ON ONE
     MOVE  ACCOUNT  TO  OUTACCT
ENDONE
CONTROL ACCOUNT
ON CHANGE  IN ACCOUNT
* CALCULATE AN AVERAGE FOR EACH DOLLAR FIELD THAT IS TOTALED

  AVG1=SUM BALANCE/TALLY ROUNDED  ;CALCULATE BALANCE AVERAGE
  AVG2=SUM INSTLBAL/TALLY ROUNDED ;CALCULATE INSTALLMENT BALANCE AVERAGE
  AVG3=SUM INSTPAY/TALLY ROUNDED  ;CALCULATE INSTALLMENT PARTIAL AVERAGE
  AVG4=SUM BALPART/TALLY ROUNDED  ;CALCULATE BALANCE PARTIAL BALANCE
  AVG5=SUM INTPART/TALLY ROUNDED  ;CALCULATE INTEREST PARTIAL AVERAGE
* CREATE OUTPUT SUMMARY RECORDS
   MOVE  SUM  BALANCE  TO  OUTBALANCE
   MOVE  SUM  INSTLBAL TO  OUTINSTAL  MOVE SUM INSTPAY  TO OUTINSTPAY
   MOVE  SUM  BALPART  TO OUTBALPART MOVE SUM INTPART TO OUTINTPART
   WRITE OUTFILE
   MOVE ACCOUNT TO OUTACCT

 LIST ACCOUNT SUM BALANCE SUM INSTLBAL SUM INSTPAY
            SUM BALPART SUM INTPART  TALLY (COUNT)
              WITH 2 BEFORE AND 2 AFTER
LIST 'AVERAGE' AT ACCOUNT AVG1 AVG2 AVG3 AVG4
                           AVG5  WITH 2 AFTER
ON FINAL
LIST SUM BALANCE SUM INSTLBAL SUM INSTPAY
     SUM BALPART SUM INTPART    WITH 2 BEFORE
T1 'VISION:RESULTS REPORT 15'  WITH 2 AFTER
T1+30 TODAYSDATE
T1+90 DYLPAGE
```

Figure 44 Creating Summary Output File and Recap by Account

The report demonstrates the results of the rounding feature (compare Figure 43 on page 67 and Figure 48 on page 71 with Figure 46 on page 69 and Figure 52 on page 73). In addition, the value of TALLY is printed at each change in ACCOUNT.

An output file containing summary records is created at control break time.

```
         1         2         3         4         5         6         7         8         9
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

VISION:RESULTS CONTROL TOTALS

FILE        RECORD          CHARACTER      BLOCK     DROPPED     REWRITTEN      INSERTED      ERASED
 ID          COUNT              COUNT      COUNT  BLOCK COUNT  RECORD COUNT  RECORD COUNT RECORD COUNT

ARFILE            200             70,400

OUTFILE            21                525


                                RECORDS                 PAGES

FILE PRINT                                                  1

REPORT PRINT                                                3

FIXED BLANK COUNT

FIXED DECIMAL DIVIDE

    RETURN CODE-0000
```

Figure 45 Summary Output File

```
                    12/01/00              VISION:RESULTS REPORT 15              PAGE        1

ACCOUNT          BALANCE          INSTALL BALANCE          INSTALL PAY          BALANCE PART PAY          INTEREST PART PAY          COUNT

  BO               44.99                    42.41                .00                       .00                       .00              1

AVERAGE            44.99                    42.41

  EO               52.00                 2,764.52              193.60                    181.98                     11.62              3

AVERAGE            17.33                   921.51               64.53                     60.66                      3.87

  FO              300.74                 1,495.07              234.76                    229.24                      5.52              7

AVERAGE            42.96                   213.58               33.54                     32.75                       .79

  IO              681.30                 2,396.96              409.12                    399.50                      9.62              6

AVERAGE           113.55                   399.49               68.19                     66.58                      1.60

  KO            3,765.92                19,441.70            1,223.76                  1,142.12                     81.64             22

AVERAGE           171.18                   883.71               55.63                     51.91                      3.71

  LO               45.24                   298.66               50.98                     49.78                      1.20              1

AVERAGE            45.24                   298.66               50.98                     49.78                      1.20
```

Figure 46 Report Showing Recap by Account (Page 1 of 2)

```
                    12/01/00          VISION:RESULTS REPORT 15          PAGE       3

ACCOUNT       BALANCE          INSTALL BALANCE          INSTALL PAY          BALANCE PART PAY          INTEREST PART PAY          COUNT

   TO          6,875.90             7,559.31               460.51                428.03                  32.48                  7
AVERAGE          982.27             1,079.90                65.79                 61.15                   4.64

   VO           41.91               100.98                17.24                 16.83                    .41                  1
AVERAGE          41.91               100.98                17.24                 16.83                    .41

   WE           11.20               268.64                45.85                 44.77                   1.08                  1
AVERAGE          11.20               268.64                45.85                 44.77                   1.08

   WO        2,034.15            17,526.50             1,499.54              1,426.59                  72.95                 28
AVERAGE          72.65               625.95                53.56                 50.95                   2.61

             19,023.10           145,346.73            10,451.64              9,841.02                 610.62
```

Figure 46 Report Showing Recap by Account (Page 2 of 2)

Printing an Entire File in Hexadecimal

The entire file is printed in its hexadecimal equivalent, but only a few records are shown.

```
OPTION HEXPRINT ARFILE
   FILE ARFILE FB 352 5280
```

or

```
FILE ARFILE FB 352 5280
HEXPRINT ARFILE
```

Figure 47 Printing an Entire File in Hexadecimal

```
        1         2         3         4         5         6         7         8         9
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
07C8006547TORRES          ERNESTO     E021589042489000000 00000000000  0007870000      TORRES,ERNESTO                1
FFCFFFFFFFFEDDDCE444444444CDDCEED4444CFFFFFFFFFFFFFFFFFF4FFFFFFFFFFF44FFFFFFFFFF54821EDDDCE6CDDCEED44
0738006547369952000000000059552360000502158904248900000000000000000000000078700006871C369952B595523600
        23444 PARK LANE           LOS ANGELES CA                %              BOBO 000000
444444444FFFFF4DCDD4DCDC4444444444DDE4CDCCDCE4CC4444444444400166000000004900049CDCD4FFFFFF0000210000
0000000002344407192031550000000000362015753520310000000000000025C0000C0049C0049C26260000000000044C0000

                  PARASO, B MD
0000000000444444444DCDCED64C4DC444444444444444444444000000000444444444400000000044444444400000000444
C000C00C0C0000000000719126B020440000000000000000000000000C0000C000000000000C0000C000000000000C0000C000

                                                    A
444444000000000444444444400000000000000000000400049440
000000000C0000C000000000000C0000C0C0C0C0C0C00049C001

07C6208657CHO PYUNG       SUH         O021789061289000000 00000000000  0006550000     *CHO PYUNG,SUH               2
FFCFFFFFFFCCD4DEEDC444444EEC44444444DFFFFFFFFFFFFFFFFFF4FFFFFFFFFFF44FFFFFFFFFF33245CCD4DEEDC6EEC444
0736208657386078457000000248000000060217890612890000000000000000000000655000085 28C386078457B248000

        33333 PALL MALL           GLENDALE CA 91206             @               EOEO 000000     @
444444444FFFFF4DCDD4DCDD4444444444CDCDCCDC4CC4FFFFF4444444440013700000000 2000020CDCD4FFFFFF0002170004
0000000003333307133041330000000007355413503109120600000000017C0000C0030C0030C56560000000000065C0046

<<      << %         MELTZER MD
400300140644444444 4DCDEECD4DC444444444444444444444400000000044444444400000000044444444400000000444
C046C00C0C0000000004533959044000000000000000000000000C0000C000000000000C0000C000000000000C0000C000

                                                    A
444444000000000444444444400000000000000000000400002440
000000000C0000C000000000000C0000C0C0C0C0C0C00030C001

07C7082509ORTIZ           DISRAELI    O082188113088000000 00000000000  9999970000     <<S.F.MEM.HOSP.              3
FFCFFFFFFFDDECE4444444444CCEDCCDC444DFFFFFFFFFFFFFFFFFF4FFFFFFFFFFF44FFFFFFFFFF02454E4C4DCD4CDED4444
0737082509693990000000000492915390006082188113088000000000000000000000999997000011082C2B6B454B8627B000

        6789 OLD MAN RD           L.A. CA 90023                       &     & EOEO 000000
444444444FFFF4DDC4DCD4DC444444444D4C44CC4FFFFF4444444444440000000000005000050CDCD4FFFFFF0002280006
0000000006789063404150940000000003B1B03109002300000000000000000C0000C0000C0000C56560000000000077C0045

%   %    %     3
6005601006444 44F444444444444444444444444444444444444400000000044444444400000000044444444400000000444
C044C01C0C000003000000000000000000000000000000000000000C0000C000000000000C0000C000000000000C0000C000
```

Figure 48 Report Printed in Hexadecimal

Printing the First 10 Records in Hexadecimal

This program creates a hexadecimal print of the first 10 records in the file. The keyword COUNT is associated with each file and contains the input or output record count for that file. COUNT must be given a unique name, as shown above, if you want to check this value in your program logic.

```
OPTION HEXPRINT ARFILE

FILE ARFILE FB 352 5280
    COUNT   RECORDCNT
IF RECORDCNT   LE   10  ACCEPT
   ELSE STOP
ENDIF
```

or

```
FILE ARFILE FB 352 5280 COUNT RECORDCNT
IF RECORDCNT LE 10 HEXPRINT ARFILE
  ACCEPT ELSE STOP ENDIF
```

Figure 49 Printing the First 10 Records in Hexadecimal

```
        1         2         3         4         5         6         7         8         9
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

07C8006547TORRES          ERNESTO     E021589042489000000 00000000000  0007870000      TORRES,ERNESTO              1
FFCFFFFFFFEDDDCE444444444CDDCEED4444CFFFFFFFFFFFFFFFFFF4FFFFFFFFFFF44FFFFFFFFFF54821EDDDCB6CDDCEED44
0738006547369952000000000595523600005021589042489000000000000000000000007870000687 1C369952B595523600

         23444 PARK LANE          LOS ANGELES CA              %               BOBO 000000
444444444FFFFF4DCDD4DCDC4444444444DDE4CDCCDCE4CC444444444440016600000004900049CDCD4FFFFFF0000210000
0000000002344407192031550000000000362015753520310000000000000025C0000C0049C0049C2626000000000044C0000

                  PARASO, B MD
00000000004444444444DCDCED64C4DC444444444444444444444000000000444444444400000000044444444400000000444
C000C00C0C0000000000719126B020440000000000000000000000000C0000C000000000000C0000C000000000000C0000C000

                                                      A
4444440000000004444444440000000000000000000040004944C
000000000C0000C000000000000C0000C0C0C0C0C0C00049C001

07C5208657CHO PYUNG      SUH        O021789061289000000 00000000000  0006550000    *CHO PYUNG,SUH              2
FFCFFFFFFFCCD4DEEDC444444EEC44444444DFFFFFFFFFFFFFFFFFF4FFFFFFFFFFF44FFFFFFFFFF33245CCD4DEEDC6EEC444
0736208657386078457000000248000000006021789061289000000000000000000000000655000085 28C386078457B248000

         33333 PALL MALL          GLENDALE CA 91206           @               BOBO 000000     @
444444444FFFFF4DCDD4DCDD4444444444CDCDCCDC4CC4FFFFF4444444440013700000002000020CDCD4FFFFFF0002170004
0000000003333307133041330000000007355413503109120600000000017C0000C0030C0030C5656000000000065C0046

       %          MELTZER MD
400300140644444444DCDEECD4DC444444444444444444444400000000444444444000000000444444444000000004 44
C046C00C0C0000000004533959044000000000000000000000000C0000C000000000000C0000C000000000000C0000C000

                                                      A
444444000000000444444444000000000000000000040002044C
000000000C0000C000000000000C0000C0C0C0C0C0C00030C001
```

Figure 50 Hexadecimal Report of the First 10 Records

```
        1         2         3         4         5         6         7         8         9
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

VISION:RESULTS CONTROL TOTALS

FILE        RECORD          CHARACTER      BLOCK      DROPPED     REWRITTEN       INSERTED       ERASED
 ID          COUNT            COUNT        COUNT    BLOCK COUNT  RECORD COUNT  RECORD COUNT RECORD COUNT

ARFILE            11            3,872


                              RECORDS                  PAGES

FILE PRINT                         10                     5

REPORT PRINT

FIXED BLANK COUNT

FIXED DECIMAL DIVIDE


    RETURN CODE-0000
```

Figure 51 Summary Output File of 10 Records

Hexprint and File Generation of Selected Records

In this example, a new file and a hexadecimal file print are created containing only those records whose account code is equal to WO.

```
OPTION HEXPRINT ARFILE
FILE ARFILE FB 352 5280
    ACCOUNT 2 182
FILE OARFILE FB 352 5280 OUTPUT FROM ARFILE
IF   ACCOUNT   EQ   'WO'  ACCEPT
   ELSE REJECT
ENDIF
```

Figure 52 Hexprint and File Generation of Selected Records

```
        1       2       3       4       5       6       7       8       9
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

07C6216617TAUTRIM       RICHARD    R090688010189000000 00000000000  9999970000     TAUTRIM,RICHARD          173
FFCFFFFFFFECEEDCD44444444DCCCCDC4444DFFFFFFFFFFFFFFFFF4FFFFFFFFFF44FFFFFFFFFF51031ECEEDCD6DCCCCDC4
07362166173143994000000009938194000090906880101890000000000000000000099999700006575C3143994B99381940

        555 LANFRANCO ST        GLENDALE CA 91205           @             WOWO 000000     *
444444444FFF4DCDCDCDCD4EE444444444CDCDCCDC4CC4FFFFF444444440020700000001000010OEDED4FFFFFF0001650006
000000000555031569153602300000000073554135031091205000000000036C0000C0040C0040C666600000000052C0026

@  -   %        MISC
70064003064444444444DCEC444444444444444444444444444444000000000444444444400000000044444444400000000444
C020C06C0C0000000004923000000000000000000000000000000000C0000C000000000000C0000C000000000000C0000C000

                                                A
444444000000000444444444000000000000000000000040010044C
000000000C0000C000000000000C0000C0C0C0C0C0C00040C001

07C6208347HAMILTON      MERDEDES   O021989061489000000 00000000000  0007870000    %HAMILTON,MERCEDE         174
FFCFFFFFFFCCDCDEDD4444444DCDCCCCE444DFFFFFFFFFFFFFFFFF4FFFFFFFFFF44FFFFFFFFFF52216CCDCDEDD6DCDCCCCC
07362083478145336500000004594545200060219890614890000000000000000000000078700005990C81493365B4593545

S       442 W 92ND ST           SUN VALLEY                  @       &    & WOWO 000000
E44444444FFF4E4FFDC4EE4444444444444EED4ECDDCE444444444444444400067000000005000050EDED4FFFFFF00002800000
200000000442060925402300000000000024505133580000000000000000099C0000C0010C0010C666600000000056C0000

                PARASO MD
000000000044444444DCDCED4DC444444444444444444444444000000004444444440000000004444444440000000044
C000C00C0C0000000007191260440000000000000000000000000000C0000C000000000000C0000C000000000000C0000C000

                                              &   A
444444000000000444444444000000000000000000000400050044C
000000000C0000C000000000000C0000C0C0C0C0C0C00010C001
```

Figure 53 Hexprint of Selected Records

```
        1         2         3         4         5         6         7         8         9
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

VISION:RESULTS CONTROL TOTALS

FILE        RECORD         CHARACTER     BLOCK     DROPPED     REWRITTEN      INSERTED      ERASED
 ID          COUNT           COUNT        COUNT   BLOCK COUNT  RECORD COUNT  RECORD COUNT RECORD COUNT

ARFILE          200           70,400

OARFILE          28            9,856


                              RECORDS                PAGES

FILE PRINT                         28                   11

REPORT PRINT

FIXED BLANK COUNT

FIXED DECIMAL DIVIDE


    RETURN CODE-0000
```

Figure 54 Summary Output File of Selected Records

Creating a Selected Records File

```
FILE ARFILE FB 352 5280
    ACCOUNT 2 182
FILE OARFILE FB 352  5280 OUTPUT FROM ARFILE
IF   ACCOUNT   EQ   'WO'   ACCEPT
   ELSE REJECT
ENDIF
```

Figure 55 Create Selected Records File

This program creates a file containing only those records whose account code is equal to WO.

```
        1         2         3         4         5         6         7         8         9
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

VISION:RESULTS CONTROL TOTALS

FILE        RECORD         CHARACTER     BLOCK     DROPPED     REWRITTEN      INSERTED      ERASED
 ID          COUNT           COUNT        COUNT   BLOCK COUNT  RECORD COUNT  RECORD COUNT RECORD COUNT

ARFILE          200           70,400

OARFILE          28            9,856


                              RECORDS                PAGES

FILE PRINT                                                1

REPORT PRINT

FIXED BLANK COUNT

FIXED DECIMAL DIVIDE


    RETURN CODE-0000
```

Figure 56 Summary Report of Selected Records Only

Correcting Records, Printing, and Creating an Updated File

In this example, blank or zero date fields are updated and written to an output file along with the rest of the file. A hexadecimal print of the updated records is produced.

```
FILE ARFILE FB 352 5280
    DATE  6  44  CH
FILE OARFILE FB 352  5280 OUTPUT FROM ARFILE
 IF DATE  EQ  SPACES   OR
   DATE  EQ  000000
   MOVE  011599  TO DATE   NEXT
   ELSE ACCEPT    ENDIF
HEXPRINT   ARFILE    ACCEPT
```

Figure 57 Correcting Records, Printing, and Creating an Updated File

```
       1       2       3       4       5       6       7       8       9
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

07C6123228SILVA          JULIAN     R010587011599000000 000000000000  9999970000    @SILVA,JULIAN              7
FFCFFFFFFFFECDEC4444444444DEDCCD44444DFFFFFFFFFFFFFFFFFFF4FFFFFFFFFFF44FFFFFFFFFF55617ECDEC6DEDCCD4444
07361232282935100000000001439150000090105870115990000000000000000000099999700002128C29351B1439150000

       33356 WIZZY DR #32       LOS ANGELES CA 90033        @               FO 000000
444444444FFFFF4ECEEE4CD47FF4444444DDE4CDCCDCE4CC4FFFFF444440000700000000000000044CD4FFFFFF0002390006
000000000333560699980490B3200000003620157535203109003300000015C0000C0000C0000C0066000000000079C0047

%   @   %        MISC
60057019064444444444DCEC4444444444444444444444444444400000000044444444400000000044444444400000000444
C046C00C0C0000000004923000000000000000000000000000000000C0000C000000000000C0000C000000000000C0000C000

                                                A
4444440000000004444444440000000000000000000040000044C
000000000C0000C000000000000C0000C0C0C0C0C0C00000C001

07C6123317VASQUEZ        IRENE      R010587011599000000 000000000000  9999970000    %VASQUEZ,IRENE             13
FFCFFFFFFFECECECE44444444CDCDC444444DFFFFFFFFFFFFFFFFFF4FFFFFFFFFFF44FFFFFFFFFF50246ECECECE6CDCDC444
07361233175127459000000009955500000090105870115990000000000000000000999997000072140C5127459B99555000

       329 W 7TH ST             CITY OF COMMERCE CA 90040    @               IO 000000
444444444FFF4E4FEC4EE4444444444444CCEE4DC4CDDDCDCC4CC4FFFFF000070000000000000044CD4FFFFFF0004600004
000000000329060738023000000000000393806603644593503109004000015C0000C0000C0000C0096000000000094C0087

       %        MISC
3002302006444444444DCEC4444444444444444444444444444400000000044444444400000000044444444400000000444
C087C00C0C0000000004923000000000000000000000000000000000C0000C000000000000C0000C000000000000C0000C000

                                                A
4444440000000004444444440000000000000000000040000044C
000000000C0000C000000000000C0000C0C0C0C0C0C00000C001
```

Figure 58 Hexprint of Corrected and Updated Records

```
        1         2         3         4         5         6         7         8         9
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

VISION:RESULTS CONTROL TOTALS

FILE       RECORD          CHARACTER     BLOCK     DROPPED     REWRITTEN      INSERTED      ERASED
 ID         COUNT            COUNT       COUNT   BLOCK COUNT  RECORD COUNT  RECORD COUNT  RECORD COUNT

ARFILE           200           70,400

QARFILE          200           70,400


                              RECORDS                 PAGES

FILE PRINT                         18                     8

REPORT PRINT

FIXED BLANK COUNT

FIXED DECIMAL DIVIDE


    RETURN CODE-0000
```

Figure 59 Summary Report of Corrected and Updated Records

Chapter 6: Structured and USERDEFAULT Mode Programming

This chapter describes how to use structured and USERDEFAULT mode programming with VISION:Results.

Structured Programming

The theory of structured programming supports the following basic structures:

- Sequence
- Selection
- Iteration

Each structure must have only one entry point and only one exit. All three structures are fully supported by VISION:Results.

Structured Option

The structured programming option of VISION:Results has been implemented for use by programmers and data processing installations that are involved in this methodology. All the generally accepted structures have been implemented in VISION:Results, and they use the traditional verbs associated with pseudo code.

Select structured mode by coding OPTION STRUCTURED in the program. Use OPTION STRUCTURED2 if you want to use the structured2 mode of operation.

Sequence

The sequence structure is made up of one or more sequentially executed commands. You can combine any of the VISION:Results commands, including PERFORM, to form a sequence structure.

You cannot use the GOTO, ACCEPT, and REJECT commands with OPTION STRUCTURED mode, but you can use them with OPTION STRUCTURED2 mode.

Selection

Use the selection structure to control processing based on a specified condition. VISION:Results has two commands to accomplish the selection:

```
IF-THEN-ELSE-ENDIF
CASE-WHEN-ELSE-ENDCASE
```

Use the IF-THEN-ELSE-ENDIF combination to control processing based on a single predicate (command). In Figure 60, each process is composed of a command.

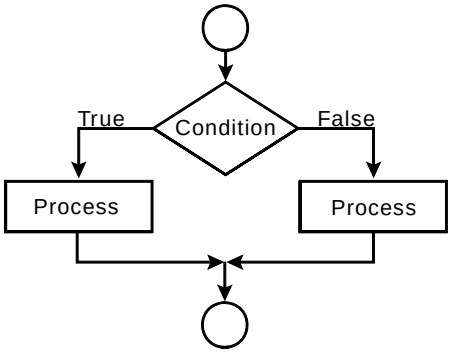

Figure 60 IF Structure

Use the CASE structure to control processing based on multiple predicates (commands).

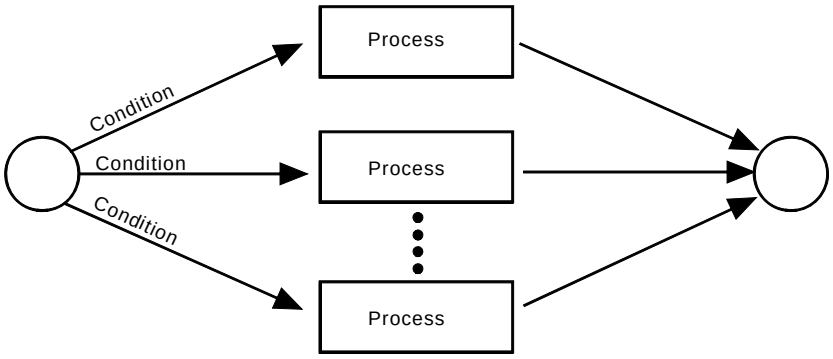

Figure 61 CASE Structure

In this structure, all predicates have the same subject but many different conditions. You can use this structure to improve readability and efficiency of the program. Suppose that processing is to be controlled based on a transaction code in the input record, and there are 50 of these codes. You could code 50 IF statements. The execution of this is inefficient and the listing is not easy to read. As an alternative, you can use the CASE structure.

Iteration

Use the iteration structure for looping, or the repeated execution of a segment of code. The commands are:

```
DOWHILE
DOUNTIL
```

The DOWHILE structure checks a specified condition and continues processing the loop while the condition is true. The condition is checked at the beginning of the loop, so it is possible to have zero iterations of the loop.

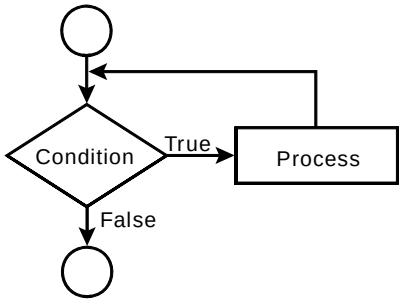

Figure 62 DOWHILE Structure

The DOUNTIL structure checks a specified condition and continues processing the loop until the condition is true. The condition is checked at the end of the loop, so the loop is always executed at least once.

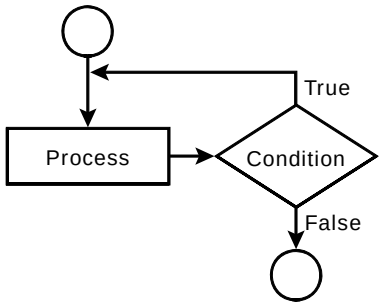

Figure 63 DOUNTIL Structure

You can combine the various structures (sequence, selection, and iteration) and nest them one within the other to accomplish any required task—the result is still a structured program. CA suggests that you indent the code for each structure that is subordinate to another to improve the readability of the resulting source listing.

USERDEFAULT Mode Programming

The USERDEFAULT mode assumes certain defaults to simplify how you use VISION:Results. For novice programmers, the USERDEFAULT is a quick and easy method to produce lists and reports.

USERDEFAULT Option

The USERDEFAULT option requires a minimum of four statements to produce a list or report. The four statements are:

- OPTION statement
- FILE statement
- Field definition statement
- LIST statement

For more information about structured and USERDEFAULT mode programming, see the *Advantage VISION:Results for z/OS Reference Guide*.

Appendix A: Definition of Terms

Commands, Keywords, and Definitions

ACCEPT	A command that causes the record currently being processed to be accepted and have its file printed if the OPTION hexprint, print, or lowercase print is specified, or to be accepted to an output file if one is being written and no WRITE commands are being used. ACCEPT does not cause a detail line to be printed unless a standard aging report is specified in VISION:Excel.
byte	In VISION:Results, a byte and a character are synonymous.
character	A character is one symbol from a set of symbols such as keys on a keyboard or the digits on a calculator. The basic symbols include the decimal digits 0 through 9, the letters A through Z, punctuation marks, and special symbols (for example, $, =, +, *, and %).
field	A field is a unit of information. A unit can be 1 byte long or many bytes long. The 1-byte field represented by the character A contains the letter A. A 2-byte field represented by any two characters contains those two characters, and so on. Generally, most fields have maximum lengths. For example, in a name field that has a maximum length of 40, the length includes space for both first and last names and as many blanks as necessary to equal 40 characters.
file	A file is a collection of one or more records. It generally resides on an external medium such as a cartridge, tape, or disk. When an automatic read command is issued by VISION:Results, one record is transferred from the external medium to the computer memory. The contents of that record are now accessible to the VISION:Results program.

IF

A command and a conditional statement where one field can be tested against another, or a field can be examined to see if its value is positive, negative, or numeric. An ENDIF keyword must be used to terminate the logic governed by the IF statement. For example:

```
IF FIELDA NUMERIC GOTO ABC ENDIF

IF FIELDA GE FIELDB GOTO DEF ENDIF
```

Valid operators are:

LT	Test field (A) for less than field (B).
GT	Test field (A) for greater than field (B).
EQ	Test field (A) for equal to field (B).
LE	Test field (A) for less than or equal to field (B).
GE	Test field (A) for greater than or equal to field (B).
NE	Test field (A) for unequal to field (B).
POSITIVE	Test for field (A) greater than zero.
NEGATIVE	Test for field (A) less than zero.
NUMERIC	Test zoned decimal field (A) for a valid numeric condition.

REJECT

A command used to reject the record currently being processed. Processing then continues with the first procedure logic statement in your program. However, if files are being read automatically, the input phase is invoked first, then the procedure logic.

record

A record is a collection of fields relating to a specific unit of information. Record length is determined by adding the length of all the fields within the record.

record formats

Used to identify the record type and if blocking is present. Valid formats are:

F	Format is fixed unblocked.
FB	Format is fixed blocked.
U	Format is undefined.
V	Format is variable unblocked.
VB	Format is variable blocked.
S	Format is variable-spanned (z/OS only).
SB	Format is variable-spanned blocked (z/OS only).

STOP

A command that terminates processing. If ON CHANGE IN and ON FINAL statements are a part of the program, VISION:Results forces all ON CHANGE IN (starting with the most minor to the most major) and ON FINAL subordinate statements to be invoked. Run statistics are written and all files are closed. Once completed, control returns to the operating system.

SUM data name	The keyword SUM is a qualifier that instructs VISION:Results to use the total of the field specified up to this point. SUM is valid during control break processing only. If you are performing detail printing, VISION:Results adds all fields that have to be totaled to appropriate accumulators whenever the first VISION:Results LIST statement is executed. If there is no detail, VISION:Results totals all required fields whenever it exits the automatic cycle.

tag name	The identification name or letter/number combination that identifies a statement or procedure used to branch from one statement to another. When used in conjunction with a GOTO, it would be specified as:
	`GOTO LISTDET`
	When used as a reference point, it should be specified with a colon (:).
	`LISTDET:`

TALLY (detail printing)	A count of the number of times the first LIST statement in the VISION:Results program is executed.
TALLY (no detail printing)	A count of the number of times that VISION:Results exited the automatic cycle. This happens when you run out of VISION:Results code to execute or you issue an ACCEPT. **Note:** TALLY is valid only during control break processing and always contains the count up to that control break. TALLY can be considered as the count of the number of records involved in a particular control break, and can be printed and written to a file or used in a calculation.

Command and Definition Statement Parameters

Use the following parameters in your command/definition statement syntax.

alphanumeric	Beginning with A to Z and containing any of the characters A to Z, 0 to 9, or $ % ! ? # ' \ [ ] { } –.
alphanumeric literal	When used to give an initial value to a data name (VALUE literal), it can be 1 to 255 characters long. When used in a TITLE statement, it can be up to the report width. (It must, however, be physically broken up into pieces contained on one free-form line.) When used in a MOVE statement, it can be 1 to 70 characters long. This type of literal begins and ends with a single or double quotation mark. If it begins with a single quotation mark and a single quotation mark is to be part of the literal, then all single quotation marks that are part of the literal must be repeated (the syntax analyzer strips out the repeated quotation marks). For example, the literal IT'S can be written as "IT'S" or 'IT''S.' The same is true of the literals beginning with a double quotation mark that contain a double quotation mark.
blocksize	A 1- to 5-character numeric size of a block. The maximum block size is 32767. The maximum variable length block is 32760.
column heading	A 1- to 30-character column heading to be printed for the data name.
data name	A 2- to 50-character alphanumeric name used in a record or work area field definition, or a VISION:Results reserved word (begins with DYL). The data name can vary up to 50 characters in length based on installation time options.
data name2	A 2- to 10-character alphanumeric name used in a FILE, TABLE, or ARRAY statement following a keyword. A self-defining data name that *must not* be defined in record or work area field definitions.
data name3	A 2- to 10-character alphanumeric name of a record or work area field definition data name that is specified in the parameter list of the keyword PARM of a MODIFY or EXIT FILE definition.
decimals	A single numeric character specifying the number of digits to the right of the assumed decimal point in the field.
file name	A 1- to 8-character (1 to 7 for VSE) alphanumeric name of a file.
hex literal	A 5- to 21-character literal that specifies a hexadecimal value. It is in the form X'nn', where nn is a pair of hex digits (a hex digit is 0 to 9 and A to F).
idname	See file name (above).
imperative	One or more commands that accomplish the purpose of the conditional selection. An imperative cannot contain another IF, except in STRUCTURED mode.

index	Either (INW), (INX), (INY), or (INZ).
length	A 1- to 5-character numeric length of a data name.
literal	See numeric literal, alphanumeric literal, and hex literal.
membername	The name of a member in the PDS specified by the file name DD statement.
modulename	A 1- to 8-character alphanumeric name of an external load module to be called.
n	A number whose attributes are determined by the command or keyword that uses it.
nK	Amount of memory required in 1K increments, where 1K is 1024 bytes; defaults to 4K.
nM	Amount of memory required in 1M increments, where 1M is 1024K.
numeric	Containing only the characters 0 to 9. (This is different from a numeric literal, which can also contain a period, commas, and a leading minus sign.)
numeric literal	A 1- to 20-character value. It can contain only the digits 0 to 9, one decimal point, commas to denote thousands, and a leading minus (–) sign.
password	A 1- to 8-character alphanumeric password (must be in single quotation marks).
printdecimals	A single numeric character specifying the number of digits to be printed to the right of the decimal point.
printdigits	1 to 2 numeric characters specifying the number of digits to be printed to the left of the decimal point. It cannot be greater than 26.
printsize	1 to 2 numeric characters specifying the print size of a numeric field using an edit code in the text of the letter writing facility.
recordsize	A 1- to 5-character numeric size of a record. The maximum record size is 32767. (The maximum variable length record is 32752.)
start	A 1- to 5-character numeric starting location of a data name within a record or work area.
tag name	A 1- to 10-character name, starting with an alphabetic or a numeric and ending with a colon (:), or a 1- to 10-digit numeric name to be associated with GOTO and PERFORM commands.
title	A 1- to n-character title can be used. However, if the title is wider than 70 characters, repeat the title specification.

```
T1 'xxxx xxxxxx xxxxx xxxx xx xxxxxxx xxxx xxxxx xx xxxx xxxxx xxxxx xxxxxx'
T1+70 'xxxxxxxxxxxxxxxxxxxxxxxxxx'
```

Arithmetic Symbols

Arithmetic operators are as follows:

*	multiplication
/	division
+	addition
-	subtraction
**	exponentiation

Examples of expressions containing arithmetic operators:

```
TOTAL=COUNT + 32

BALANCE = BALANCE * 1.10

JBALANCE=BALANCE*1.10

ADJBUD=YTDBUDGET*(INFLA/1.12)

TOTAL=(AVG+10)*((BAL/8)=20)

SQUARE=5**2
```

VISION:Results performs all computations according to the rules of algebra. Each of the operators is assigned a priority, as follows:

Exponentiation (**)	Priority 1
Multiplication and division (*,/)	Priority 2
Addition and subtraction (+,-)	Priority 3

This priority is used to determine in what order to perform the operations. Priority 1 is performed ahead of priority 2, and priority 2 is performed ahead of priority 3. Operations with the same priority are performed in the order in which they are coded, except for exponentiation. Sequential exponentiation operators are processed from right to left.

Data Types

Data types are as follows:

NU	Numeric data in zoned decimal format. Numeric values from 0 to 9.
CH	Alphabetic or alphanumeric data; this is the default taken by VISION:Results and need not be specified. Values A through Z and any combination of numeric and special characters.
PD	Packed numeric data (two digits per byte).
BI	Binary numeric value.

Edit Codes

You can supply any of the following edit codes in your field definition to override the default edit format (P edit code).

Edit Code	Description	Examples
E	Edit with zero suppression, commas, decimal insertion, and negative sign to the right of the field. Blank if zero.	001234.56 prints as 1,234.56 000000.00 prints as (blank)
Y	Leading '-'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with zero suppression, commas, decimal insertion, and negative sign to the left of the number. Blank if zero.	-001234.56 prints as -1,234.56 000000.00 prints as (blank)
NE	Leading '-'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with zero suppression, commas, decimal insertion, and negative sign to the left of the number. Blank if zero.	-001234.56 prints as -1,234.56 000000.00 prints as (blank)
DE	Leading '$'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with zero suppression, commas, decimal insertion, and negative sign to the right of the field. Float a currency sign (the default is $) to the left of the most significant digit. Blank if zero.	-001234.56 prints as $1,234.56- 000000.00 prints as (blank)
F	Leading '$'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with zero suppression, commas, decimal insertion, and negative sign to the right of the field. Float a currency sign to the left of the field. Blank if zero.	-001234.56 prints as $1,234.56- 000000.00 prints as (blank)
NDE	Leading '-$'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with zero suppression, commas, decimal insertion, and negative sign to the left of the field. Float a currency sign to the direct left of the field. Blank if zero.	-001234.56 prints as -$1,234.56 000000.00 prints as (blank)
G	Leading '-$'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with zero suppression, commas, decimal insertion, and negative sign to the left of the field. Float a currency sign to the direct left of the field. Blank if zero.	-001234.56 prints as -$1,234.56 000000.00 prints as (blank)
DNE	Leading '$-'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with zero suppression, commas, decimal insertion, and negative sign to the direct left of the field. Float a currency sign to the left of the negative sign. Blank if zero.	-001234.56 prints as $-1,234.56 00000000 prints as (blank)
K	Leading '$-'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with zero suppression, commas, decimal insertion, and negative sign to the direct left of the field. Float a currency sign to the left of the negative sign. Blank if zero.	-001234.56 prints as $-1,234.56 00000000 prints as (blank)

Edit Code	Description	Examples
A	Edit with zero suppression, commas, decimal insertion, and negative sign to the right of the field. Print decimal point and zeros to the right of decimal. If field has no decimal positions and has a zero value, nothing prints.	001234.56 prints as 1,234.56 0000000000 prints as (blank)
X	Leading '-'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with zero suppression, commas, decimal insertion, and negative sign to the left of the field. Print decimal point and zeros to the right of decimal. If field has no decimal positions and has a zero value, nothing prints.	001234.56 prints as 1,234.56 -001234.56 prints as -1,234.56 0000000.00 prints as .00 0000000000 prints as (blank)
NA	Leading '-'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with zero suppression, commas, decimal insertion, and negative sign to the left of the field. Print decimal point and zeros to the right of decimal. If field has no decimal positions and has a zero value, nothing prints.	001234.56 prints as 1,234.56 -001234.56 prints as -1,234.56 0000000.00 prints as .00 0000000000 prints as (blank)
DA	Leading '$'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with zero suppression, commas, decimal insertion, and negative sign to the right of the field. Float a currency sign (the default is $) to the left of the most significant digit. Blank if zero.	001234.56 prints as $1,234.56 -001234.56 prints as $1,234.56- 0000000.00 prints as $.00 0000000000 prints as (blank)
H	Leading '$'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with zero suppression, commas, decimal insertion, and negative sign to the right of the field. Float a currency sign (the default is $) to the left of the most significant digit. Blank if zero.	001234.56 prints as $1,234.56 -001234.56 prints as $1,234.56- 0000000.00 prints as $.00 0000000000 prints as (blank)
NDA	Leading '-$'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with zero suppression, commas, decimal insertion, and negative sign to the left of the field. Float a currency sign to the direct left of the field. Blank if zero.	001234.56 prints as $1,234.56 -001234.56 prints as -$1,234.56 0000000.00 prints as $.00 0000000000 prints as (blank)
J	Leading '-$'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with zero suppression, commas, decimal insertion, and negative sign to the left of the field. Float a currency sign to the direct left of the field. Blank if zero.	001234.56 prints as $1,234.56 -001234.56 prints as -$1,234.56 0000000.00 prints as $.00 0000000000 prints as (blank)
DNA	Leading '$-'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with zero suppression, commas, decimal insertion, and negative sign to the direct left of the field. Float a currency sign to the left of the negative sign. Blank if zero.	001234.56 prints as $1,234.56 -001234.56 prints as $-1,234.56 0000.00 prints as $.00 0000000000 prints as (blank)

Edit Code	Description	Examples
L	Leading '$-'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with zero suppression, commas, decimal insertion, and negative sign to the direct left of the field. Float a currency sign to the left of the negative sign. Blank if zero.	001234.56 prints as $1,234.56 -001234.56 prints as $-1,234.56 0000.00 prints as $.00 0000000000 prints as (blank)
Z	Edit with zero suppression, decimal insertion, and negative sign to the right of the field, but no commas. Blank if zero.	001234.56 prints as 1234.56 00000000 prints as (blank)
U	Leading '-'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with zero suppression, decimal insertion, and negative sign to the left of the field, but no commas. Blank if zero.	00123456 prints as 1234.56 -00123456 prints as -1234.56 00000000 prints as (blank)
NZ	Leading '-'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with zero suppression, decimal insertion, and negative sign to the left of the field, but no commas. Blank if zero.	001234.56 prints as 1234.56 -001234.56 prints as -1234.56 00000000 prints as (blank)
B	Edit with zero suppression, decimal insertion, and negative sign to the right of the field, but no commas. Blank if zero. If zero value, print decimal point and zeros to the right of decimal. If field has no decimal positions and has a zero value, nothing prints.	001234.56 prints as 1234.56 000000.00 prints as .00 000000000 prints as (blank)
Q	Leading '-'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with zero suppression, decimal insertion, and negative sign to the left of the field, but no commas. Blank if zero. If zero value, print decimal point and zeros to the right of decimal. If field has no decimal positions and has a zero value, nothing prints.	001234.56 prints as 1234.56 -001234.56 prints as -1234.56 000000.00 prints as .00 000000000 prints as (blank)
NB	Leading '-'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with zero suppression, decimal insertion, and negative sign to the left of the field, but no commas. Blank if zero. If zero value, print decimal point and zeros to the right of decimal. If field has no decimal positions and has a zero value, nothing prints.	001234.56 prints as 1234.56 -001234.56 prints as -1234.56 000000.00 prints as .00 000000000 prints as (blank)
P	Default edit code. Edit with decimal insertion, and negative sign to the right of the field, but no commas. If the field has no decimal positions and has a zero value, zeros print. If it has a zero value, zeros print.	001234.56 prints as 001234.56 000000000 prints as 000000000
W	Leading '-'. For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit with decimal insertion, and negative sign to the left of the field, but no commas. If the field has no decimal positions and has a zero value, zeros print. If it has a zero value, zeros print.	-001234.56 prints as -001234.56 0000000.00 prints as 0000000.00 000000000 prints as 000000000

Edit Code	Description	Examples
NP	Leading '-'. Applies only to data name definitions and LIST and EDIT statement overrides. Edit with decimal insertion and negative sign to the left of the field, but no commas. If the field has no decimal positions and has a zero value, zeros print. If it has a zero value, zeros print.	-001234.56 prints as -001234.56 0000000.00 prints as 0000000.00 000000000 prints as 000000000
D	Date edit of a 6-byte zoned decimal or 4-byte packed field.	120100 prints as 12/01/00
V	For use in data name definitions, LIST, EDIT, and LTD (Letterwriter) statements. Edit a date field containing a 4-digit year. The date field must be either an 8-byte zoned decimal or 5-byte packed field.	12012000 prints as 12/01/2000
D4	Applies only to data name definitions and LIST and EDIT statement overrides. Edit a date field containing a 4-digit year. The date field must be either an 8-byte zoned decimal or 5-byte packed field.	12012000 prints as 12/01/2000
S	Edit a social security number from a 9-byte numeric or 5-byte packed decimal field.	999999999 prints as 999-99-9999

Appendix B: Numeric and Packed Key Considerations

When you process VSAM files, while building records or retrieving records in random or skip mode, consider the key sign of the key if the key has a data type of numeric (zoned decimal) or packed.

A positive sign for a numeric or packed field is C or F. This can be a problem if you are searching for a particular record on a VSAM file, and the key you are using to search with has a different sign than the key in the record. For example, a 2-byte packed field with the value 123C is not considered equal to a 2-byte packed field with the value 123F and VISION:Results sets a NO RECORD FOUND status on this field.

In a VISION:Results program, you might encounter an unlikely sign problem if:

- You have set up a packed key in the VSAM key area.
- This key has been changed to a C sign (perhaps the key had been converted from numeric to packed and a C sign resulted from this process).

If the key in the record contains an F sign, a status code of blank (no record found) is returned to the program after a random retrieval read. The same situation can occur with numeric fields.

To prevent or correct for unlike signs, use the following:

- Whenever possible, treat the keys as character data.

```
FILE VSAMIN KSDS F 120 RANDOM SEARCHKEY
  STATUS STATUSFLAG  KEYLEN 3
FILE TRANIN STATUS TRANEOF
  KEYIN  3  CH   ; (key is actually packed decimal)
PROCEDURE:
  IF TRANEOF EQ 'E' STOP ENDIF
  MOVE KEYIN TO SEARCHKEY    ; (sign remains unchanged)
  READ VSAMIN
  IF STATUSFLAG NE 'Y'
    GOTO NOTFOUND    ENDIF
```

In the example above, the keys actually have a data type of packed, but are defined as character, so no sign change can occur. If the key has to be treated as packed for other purposes, it can be redefined.

```
FILE VSAMIN2 KSDS F 130 RANDOM FINDKEY
  STATUS VSSTAT KEYLEN 5
  ACCOUNT 5  NU ...
FILE TRANSIN  STATUS  TRANSTAT
  TRANACCT 5 ...
PROCEDURE:
  IF TRANSTAT EQ 'E' STOP ENDIF
  MOVE TRANACCT TO FINDKEY
  READ VSAMIN2 .
```

In the example above, a numeric key in the transaction file (TRANSIN) is defined as character. FINDKEY is also defined as character (default). No sign change happens when TRANACCT is moved to FINDKEY.

- If you need to convert a packed key sign from C to F, you can apply OR to the field.

```
COMBINE BITS X'000F' OR SRCHKEY
```

In the example above, a 2-byte packed key is having its sign set to F.

- If you need to convert a packed key's sign from F to C, you can apply AND to the field.

```
COMBINE BITS X'FFFC' AND SRCHKEY
```

In the example above, a 2-byte packed key with an F sign is having its sign set to C.

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z