

CHAPTER 10

GENERATING AN INTAC REPORT

GENERATE AN INTAC REPORT

INTAC offers two facilities for defining reports: GENERATE and INQUIRE. INQUIRE can be used interactively to produce relatively simple reports quickly and easily (see Chapter 9). GENERATE, which initially requires a three-step process, offers greater calculation power and more flexible report formatting.

To use GENERATE, you create a report definition which may customize the format, sort records, print your choice of data items, and even calculate new data items. Once you have created this "report definition file" (step 1) and used GENERATE to create a program (step 2), you can produce the report time after time by simply running the program (step 3).

You can use up to three files to prepare reports, one primary file and one or two reference files. For example, an employee master file can be sorted and printed, and a secondary department file can be used to look up the department name for each employee. Defined data items can appear on the report and be used to link the primary file to secondary files. The defined items may be calculated or prompted values. To simplify common calculations, INTAC has many built-in functions for use in your GE programs.

The following report shows some of the flexible formatting possible with the Report Program Generator.

DATE: 03-Jul-82

TRI-CITY COMPANY

DEPARTMENT - 1001 ACCOUNTING
MONTHLY DEPRECIATION ON ALL PURCHASED ITEMS
STRAIGHT LINE METHOD

ASSET NO	DESCRIPTION	ASSET TYPE	ORIG.COST	LIFE	BEG MO	DTE YR	END MO	DATE YR	MONTHLY DEPR
00002	DESK 4'	F	357.25	6	5	71	4	77	4.96
00015	H-P CALCULATOR	E	48.56	8	8	71	7	79	0.51
00016	STORAGE SHELF 4'	F	60.25	10	8	71	7	81	0.50
00017	MONTES BOOKCASE 8'	F	80.42	10	8	71	7	81	0.67
00018	SECRETARY DESK 5'	F	452.99	10	8	71	7	81	3.77
00019	BOOKCASE 60"	F	65.74	10	10	71	9	81	0.55
00020	TYPING CHAIR	F	128.95	10	10	71	9	81	1.07
00022	WRITER'S DESK, 5'	F	409.81	10	10	71	9	81	3.42
00023	STORAGE FILE 4-DR	F	107.04	10	10	71	9	81	0.89
00024	TAPEWRITER	E	615.90	10	10	71	9	81	5.13
00026	CONFERENCE TABLE	F	759.34	10	10	71	9	81	6.33
TOTAL PURCHASES FOR YEAR 71			3,086.25						
00051	LATERAL FILE	F	219.00	10	1	72	12	81	1.83
00056	RECEPTION SOFA	F	750.00	3	1	72	12	74	20.83
TOTAL PURCHASES FOR YEAR 72			969.00						
TOTAL PURCHASES			4,055.25						

GE

HOW TO USE GENERATE

Use the GENERATE facility to produce reports by following these three steps:

1. Create a report definition.

Using a text editor (such as the Ross Editor), create a report definition file with .DEF as the extension. This file contains all the information needed to create the report. Save the file with line numbers for use in the next step.

2. Generate a report program.

This step translates the report definition file into a BASIC program. INTAC validates the information in the definition file and then merges it with a partial program called a "skeleton" to form the complete program. INTAC offers a choice of two different skeletons depending on your needs and the computer system you are using.

To initiate this step, enter INTAC, give the GE command, and follow the GE dialogue as shown later in this chapter. You only need to perform steps 1 and 2 one time. (Of course, you may make changes to the report definition file and then generate a new program.)

3. Running the report.

Whenever you wish to run the report, you can now simply run the BASIC program produced in step 2, using the appropriate system run command.

This chapter is organized to reflect these three steps. First, the chapter explains how to plan the report and create the report definition. Each section of the definition is explained in detail. "Report Program Generation" beginning on page 10-35 explains step 2. "Obtaining Reports" beginning on page 10-38 explains step 3, including several different ways by which you can print the report. Additional technical information for experienced programmers can be found in Appendixes B and C.

THE REPORT DEFINITION FILE

Create the report definition file with a text editor. Give the file a name using .DEF as the extension. Save the report definition file with line numbers. Type in the filename in response to prompts in the INTAC GENERATE facility when you create the report program (step 2). The report program will have the same filename but a different extension. You will use the filename with the RUN command at the system level to produce the report (step 3).

This part of the chapter discusses topics of general relevance for producing the report definition file. Each section of the definition file is described in detail.

TOPICS

- Planning the GE Report
- Report Definition File Example
- Sections of a Report Definition File
- Line Numbers and Comments
- Data Item Names
- Control Breaks
- Section Descriptions

PLANNING THE GE REPORT

The complexity of reporting possible with the GE facility is virtually unlimited. An experienced user can use the facility and built-in INTAC functions to perform any calculation. At the same time, even a beginner can follow the examples in this chapter to create sophisticated reports.

As you plan your report, it is handy to have listings of the file definitions you will be using. The listing (obtained with the LI command) shows data items, their print formats, and the indexes that can be used to link the files. (You can also define new data items to use as indexes.)

Plan your report by preparing a rough sketch. Note on your sketch any additional data items you will want to define to receive values from prompting or from calculations. Also write down any necessary calculations. If you are an experienced programmer, you will want to check Appendixes B and C for technical information such as using BASIC code, in-core tables, and built-in functions for common calculations.

At the end of this chapter is a worksheet that can be reproduced for use as you plan GE file definitions. The worksheet calls for comments to make the file self-explanatory for future reference. Using this worksheet and the instructions in this chapter, write down your specifications for each section of the GE program. We recommend that you plan these sections in this order: SELECT, COLUMNS, DEFINE, TITLE, LOGIC, CODE, OPTIONS. Now you are ready to create the report definition file using a system text editor.

REPORT DEFINITION FILE EXAMPLE

Below is an example of a report definition file and part of the report it defines. The purposes of all sections of the definition are explained on the following page.

```
11 !*****
12 !GENERATE DEFINITION FILE NAME: DEPR.DEF
13 !PURPOSE OF FILE:  MONTHLY DEPRECIATION REPORT
14 !AUTHOR:  S. HOPE
15 !DATE CREATED:  26-APR-82
16 !*****
17 !
100 OPTIONS
110      FORM
```

```

120      WIDTH 80
130      NEGATIVE L
200 DEFINE                                !GOOD HABIT TO DEFINE LOGIC VARIABLES
210      END.MO I 3
220      END.YR I 3
230      MONTHLY.DEPR R 10.2
300 SELECT
310      FILE1 ASSET
320      INCLUDE IF ASSET.CODE = "P"
330      FILE2 DEPT INDEX1 DEPT.NO
340      SORT DEPT.NO PUR.YEAR PUR.MONTH ASSET.NO
350      BREAK DEPT.NO PUR.YEAR
400 TITLE
410      LEFT DATE
420      CENT "TRI-CITY COMPANY"
430      LEFT " "
440      CENT "DEPARTMENT - " DEPT.NO " " DEPT.NAME
450      CENT "MONTHLY DEPRECIATION ON ALL PURCHASED ITEMS"
460      CENT "STRAIGHT LINE METHOD"
470      LEFT " "
500 COLUMNS U-
510      ASSET.NO:COL1
520      DESCRIPTION:COL2
530      ASSET.TYPE:COL3
540      ORIG.COST:COL4
550      LIFE:COL5
560      PUR.MONTH "BEG/MO"
570      PUR.YEAR "DTE/YR"
580      "END/MO" 3
590      "DATE/YR" 4
600      "MONTHLY/DEPR" 10.2
3500 LOGIC
5000      BEGIN REPORT                                !BREAKO
6000      BEGIN DEPT.NO                                !BREAK1
6010      NEW-PAGE
12000     DETAIL PROCESSING
12010      MONTHLY.DEPR = ORIG.COST/(LIFE * 12)
12020      END.YR = PUR.YEAR + LIFE
12030      END.MO = PUR.MONTH - 1
12040      IF END.MO = 0 THEN &
              END.MO = 12 &
              \END.YR = END.YR - 1
12050      DISPLAY ASSET.NO:COL1 DESCRIPTION ASSET.TYPE ORIG.COST LIFE +
12060      DISPLAY PUR.MONTH:COL6 PUR.YEAR END.MO END.YR MONTHLY.DEPR
12070      TOTAL ORIG.COST
13000     END PUR.YEAR                                !BREAK2
13010      DISPLAY "-----":COL4
13020      DISPLAY "TOTAL PURCHASES FOR YEAR ":5 PUR.YEAR:30 T2.ORIG.COST:COL4
13030      SKIP1
14000     END DEPT.NO                                !BREAK1
14010      DISPLAY "-----":COL4
14020      DISPLAY "TOTAL PURCHASES .....":5 T1.ORIG.COST:COL4
15000     END REPORT                                !BREAKO

```

DATE: 03-JUL-82

TRI-CITY COMPANY

DEPARTMENT - 6501 PLANNING
MONTHLY DEPRECIATION ON ALL PURCHASED ITEMS
STRAIGHT LINE METHOD

ASSET NO	DESCRIPTION	ASSET TYPE	ORIG.COST	LIFE	BEG MO	DTE YR	END MO	DATE YR	MONTHLY DEPR
00008	PROJECTOR	E	256.10	5	10	71	9	76	4.27
00032	CRYSTAL BALL	E	55.00	10	11	71	10	81	0.46
TOTAL PURCHASES FOR YEAR 71			311.10						
00048	DRAFTING TABLE	F	329.00	5	1	72	12	76	5.48
00049	DRAFTING CHAIR	F	69.00	5	1	72	12	76	1.15
00050	ROUND CONF TABLE	F	109.00	5	1	72	12	76	1.82
TOTAL PURCHASES FOR YEAR 72			507.00						
TOTAL PURCHASES			818.10						

SECTIONS OF A REPORT DEFINITION FILE

Every INTAC report definition file has several parts or sections arranged in a fixed order. There are a maximum of seven sections, but only the SELECT, TITLE, and LOGIC sections are required. Each section begins with the name of the section and is followed by information appropriate to that section. Line numbers for sections must keep the file in the order shown in the chart below:

SECTION NAME	DESCRIPTION OF PURPOSE	PAGE
OPTIONS	Defines various processing and printing options, such as page width and treatment of negative numbers. Specifies use of either the standard or the expanded program skeleton (see DEFINE and SELECT sections).	10-13
DEFINE	Defines temporary data items used by the report; these are data items not found on any of the INTAC files used by the report. These items may hold the results of calculations or may receive values through prompting. Use in titles or columns headings.	10-17
SELECT	Defines up to 3 INTAC files to be used by the report program; also indicates the records to be selected and the sort sequence and control break fields on the report. (Use of some SELECT statements requires use of the non-standard program skeleton in the OPTIONS section.)	10-19
TITLE	Defines the report headings on the top of each page.	10-25
COLUMNS	Defines columns of data, column headings, and print format of the	

data in the columns on the report.

10-26

LOGIC Defines printing and calculation rules for the various parts of the report. Can include INTAC functions (see Appendix C for available functions).

10-28

CODE Reserved for user BASIC subroutines and functions. (See Appendix C for available functions.)

10-33

As you write your definition file, we recommend that you create these sections in an order consistent with your planning process: **SELECT**, **COLUMNS**, **DEFINE**, **TITLE**, **LOGIC**, **CODE**, **OPTIONS**. Assign line numbers to keep the file in the order necessary for the GE facility.

LINE NUMBERS AND COMMENTS

LINE NUMBER RULES

Each line in a report definition file has a line number followed by the required information. The line numbers usually have no significance except to keep the file in the order necessary for the GE facility. (In the CODE section, the use of specific line numbers is required.) Ross Systems recommends that you begin sections with line numbers as indicated throughout this chapter to make the file easy to read. Save the report definition file with line numbers.

MULTIPLE STATEMENTS ON A LINE

INTAC allows you to place more than one statement on a line in a GE program. A "line" is the code associated with a line number. A "subline" is a physical line separated from the previous physical line by an ampersand, carriage-return. A "statement" may be an INTAC or BASIC statement.

Each INTAC or BASIC statement should be on its own subline. Use a backslash (\) at the beginning of a subline, between multiple statements on one line.

INTAC does not allow statement modifiers after INTAC statements. (That is, you cannot have a statement such as DISPLAY DESCRIPTION IF DEPT.NO=2001)

CORRECT EXAMPLE:

```
5030 IF DEPT.NO >4001 THEN &  
      DEPT.TOTAL=DEPT.TOTAL + SALES.TOTAL &  
      \DISPLAY DEPT.TOTAL:COL5
```

GE

COMMENTS

Comments may be included in a report definition file by preceding the comment with an exclamation point (!). Everything on the line following an exclamation point is considered to be a comment, and is ignored by the report generator. The comment is best placed on a line by itself. A comment may, however, be placed at the end of a line after most report generator statements.

DATA ITEM NAMES

A report definition file may refer in various sections to data item names. The data item names may be either items in the primary or secondary files or defined items.

Each data item name consists of from 1 to 24 alphabetic and numeric characters and periods. The first character must be alphabetic. Names which are INTAC or BASIC reserved words may not be used. (See Appendix D for a list of reserved words.) If a data item name contains a space, it must be replaced by a period within a report definition file. (It is a better practice not to create data item names with embedded spaces.)

QUALIFIED NAMES

If the same name is used in more than one file and/or the DEFINE section, INTAC must be able to distinguish one from another. This is done by preceding a data item name with Fn., where n is defined as shown:

- F1 for items from FILE1,
- F2 for items from FILE2,
- F3 for items from FILE3,
- F4 for items from the report DEFINE section.

Thus, if the field DEPT.NO is defined in FILE1 and FILE3, it should be referred to as F1.DEPT.NO and F3.DEPT.NO, respectively.

QUALIFIED ITEM NUMBERS

The report definition may also refer to data items by their item numbers in the INTAC definition (or order of occurrence in the DEFINE section).

If PO.NUMBER is data item 4 in FILE1, it may be called

PO.NUMBER
F1.PO.NUMBER
F1.4

Likewise, the field DEPT which is the seventh data item in a DEFINE section, may be called

DEPT
F4.DEPT
F4.7

CONTROL BREAKS

Report control breaks occur when the value of a specified data item changes on a report. For example, if a report is to print detail records for each item in inventory, subtotalling purchase cost by department, then a control break will occur each time the department changes. After the last record for a department, the subtotals will be available to print. Typically, statements would be included in the LOGIC statement to collect these subtotals and to display them along with an appropriate heading at the end of a break.

ILLUSTRATION

Below for your comparison are

- representative data from two files: ASSET and DEPT
- part of a report (one department's assets)
- the SELECT and LOGIC sections of the report definition file that produced the report.

The Data

SEQUENTIAL DUMP OF ASSET.INT (*partial*)

DEPT NO	ASSET NO	DESCRIPTION	ASSET TYPE	ORIG.COST	PUR YR	PUR MON	LIFE
5001	00031	FLIP CHART STAND	E	45.87	71	11	10
6501	00032	CRYSTAL BALL	E	55.00	71	11	10
6001	00033	WALKIE-TALKIE	E	68.43	71	11	5
7001	00034	SECRETARY DESK 5'	F	1,096.52	71	11	10
7501	00001	CONFERENCE DESK	F	420.00	71	9	10

SEQUENTIAL DUMP OF DEPT.INT (partial)

DEPT NO	DEPT NAME
5001	ADVERTISING
6001	SECURITY
6501	PLANNING
7001	RESEARCH

SELECT and LOGIC Statements

```

300 SELECT
310     FILE1 ASSET
320     INCLUDE IF ASSET.CODE = "P"
330     FILE2 DEPT INDEX1 DEPT.NO
340     SORT DEPT.NO PUR.YEAR PUR.MONTH ASSET.NO
350     BREAK DEPT.NO PUR.YEAR

3500 LOGIC
5000     BEGIN REPORT                                !BREAK0
6000     BEGIN DEPT.NO                                !BREAK1
6010     NEW-PAGE
12000    DETAIL PROCESSING
12010        MONTHLY.DEPR = ORIG.COST/(LIFE * 12)
12020        END.YR = PUR.YEAR + LIFE
12030        END.MO = PUR.MONTH - 1
12040        IF END.MO = 0 THEN &
                END.MO = 12 &
                \END.YR = END.YR - 1
12050        DISPLAY ASSET.NO:COL1 DESCRIPTION ASSET.TYPE ORIG.COST LIFE +
12060        DISPLAY PUR.MONTH:COL6 PUR.YEAR END.MO END.YR MONTHLY.DEPR
12070        TOTAL ORIG.COST
13000    END PUR.YEAR                                !BREAK2
13010        DISPLAY "-----":COL4
13020        DISPLAY "TOTAL PURCHASES FOR YEAR ":5 PUR.YEAR:30 T2.ORIG.COST:COL4
13030        SKIP1
14000    END DEPT.NO                                !BREAK1
14010        DISPLAY "-----":COL4
14020        DISPLAY "TOTAL PURCHASES .....":5 T1.ORIG.COST:COL4
15000    END REPORT                                !BREAK0

```

Partial Report Created by File Definition

DATE: 03-JUL-82

TRI-CITY COMPANY

DEPARTMENT - 6001 SECURITY
MONTHLY DEPRECIATION ON ALL PURCHASED ITEMS
STRAIGHT LINE METHOD

ASSET NO	DESCRIPTION	ASSET TYPE	ORIG.COST	LIFE	BEG MO	DTE YR	END MO	DATE YR	MONTHLY DEPR
00007	3 CHAIRS	F	146.87	5	5	71	4	76	2.45
00100	CHAIR	F	150.00	10	5	71	4	81	1.25
00200	DESK	F	1,000.00	5	5	71	4	76	16.67
00205	TABLES	F	5,000.00	2	5	71	4	73	208.33
00044	TYPING TABLE	F	56.67	5	8	71	7	76	0.94
00033	WALKIE-TALKIE	E	68.43	5	11	71	10	76	1.14
TOTAL PURCHASES FOR YEAR 71			6,421.97						
TOTAL PURCHASES			6,421.97						

DATE: 03-JUL-82

TRI-CITY COMPANY

DEPARTMENT - 6501 PLANNING
MONTHLY DEPRECIATION ON ALL PURCHASED ITEMS
STRAIGHT LINE METHOD

ASSET NO	DESCRIPTION	ASSET TYPE	ORIG.COST	LIFE	BEG MO	DTE YR	END MO	DATE YR	MONTHLY DEPR
00008	PROJECTOR	E	256.10	5	10	71	9	76	4.27
00032	CRYSTAL BALL	E	55.00	10	11	71	10	81	0.46
TOTAL PURCHASES FOR YEAR 71			311.10						
00048	DRAFTING TABLE	F	329.00	5	1	72	12	76	5.48
00049	DRAFTING CHAIR	F	69.00	5	1	72	12	76	1.15
00050	ROUND CONF TABLE	F	109.00	5	1	72	12	76	1.82
TOTAL PURCHASES FOR YEAR 72			507.00						
TOTAL PURCHASES			818.10						

GE

CONTROL BREAK FORMATTING AND PROCESSING

You will specify control breaks on a report at places where you want certain formatting or processing to take place.

For example, at the beginning of a new group defined by a control break, you may want to give instructions to cause group names to print, or the report to skip to a new page. At the end of a group, you may want to print group totals and clear the totals before processing the next group.

The printing of a report thus involves processing at these points, relative to the control breaks:

- processing that occurs at the beginning of the various control breaks
- detail processing within the group defined by the control break
- processing that occurs at the end of the control breaks

For any given group, you might want either beginning or end processing or both beginning and end processing. Detail processing is required by GE. In the LOGIC section, you use the statements BEGIN and END to outline each break and the statement DETAIL PROCESSING.

MAJOR AND MINOR BREAKS

Control breaks are always defined from major to minor break fields. In the example, the major break is DEPT.NO and the minor break PUR.YEAR. The highest level break on every report is always the TOTAL REPORT level, and is called BREAK0. The others are numbered starting with BREAK1 for the major break. An INTAC report may have a maximum of 8 control breaks defined in addition to the TOTAL REPORT, BREAK0. (Note that the SORT sequence must be the same as the BREAK sequence.)

In the LOGIC section, use BEGIN and END statements to outline breaks for which you want special processing. In the example, the file definition results in the processing traced below. (Note that you need to include a BEGIN or END statement for any given break only if some special processing is desired at that break. The DETAIL PROCESSING statement is required.)

BEGIN REPORT !BREAK0

Report headings print and report is initialized. (The user did not really need to include a specification for this break in the file definition because nothing special was desired.)

BEGIN DEPT.NO !BREAK1

Print DEPT.NO headings and begin each DEPT.NO on a new page.

DETAIL PROCESSING

New data items are defined as the results of calculations. After the command DISPLAY the data items to print are listed. The TOTAL command causes ORIG.COST totals to be collected for three break levels and stored in the new data items T2.ORIG.COST, T1.ORIG.COST, and T0.ORIG.COST

END PUR.YEAR !BREAK2

At the end of each purchase year, INTAC prints the data item T2.ORIG.COST and the specified heading (note inclusion of a dotted line).

END DEPT.NO !BREAK1

At the end of each department, DEPT.NO totals (data item T1.ORIG.COST) and the specified heading print.

END REPORT !BREAKO

The user did not need to include this END statement since no processing is indicated for the end of the report.

OPTIONS SECTION

The OPTIONS section is used to define various processing and printing options, such as page width and treatment of negative numbers. You also specify the standard or expanded program skeleton, depending upon the complexity of your SELECT section and the number of DEFINE data items. The VMS operating system always requires (and defaults to) the expanded skeleton.

Each option has a keyword, which is sometimes followed by one or more values indicating your choice. Most options have a default as shown on the chart of options. The option must be specified only if the default is not appropriate.

GE

CHART OF OPTIONS

KEYWORD	VALUES	DEFAULT	MEANING
CHAIN	filename	none	Program to chain to at end of report
COMMA NOCOMMA		COMMA	Print commas in numbers
EDIT NOEDIT		NOEDIT (RSTS) EDIT (VMS)	Use edit parameters from file
FORM NOFORM		FORM	Form feed for page ejecting
LINES	n	66	Lines per page
MARGINB	n	3	Bottom margin
MARGINL	n	0	Left margin
MARGINI	n	3	Top margin
NEGATIVE	L, R, or P	L	Format of negative numbers
PAGE NOPAGE		PAGE	Automatic line counting and page breaks
SEGMENT	filename	defname.BAS	Name of generated report program
SKELETON	INT:INTGES INT:INTGNS	standard (RSTS) expanded (VMS)	Name of skeleton program to be used in generating report
WIDTH	n	132	Page width in characters

OPTIONS DESCRIPTIONS

CHAIN filename

Specifies the name of a program to be chained to after the report program has completed.
Default: None

COMMA (NOCOMMA)

Indicates whether or not to print commas in real numbers greater than 999 (type R items only).
Default: COMMA

EDIT (NOEDIT)

Allows use of edit parameters from the data file in the generated program. If EDIT is specified, then the report program is generated in such a way that user code may be added to access INTAC FILE1 edit parameters. NOEDIT is not available on VMS systems.
Default: NOEDIT (RSTS/E)
EDIT (VMS)

FORM (NOFORM)

Indicates whether or not to use the form-feed character (CONTROL L) to perform page ejects. FORM should be used for reports being printed on the line printer or on any terminal that has form-feed capability. If NOFORM is used, INTAC prints the number of blank lines needed to position page headings at the top of the next page using the number of lines per page specified in the LINES option.

Default: FORM

LINES n

Indicates number of lines per page. For continuous printing with no paging, specify a number larger than the number of lines on the report (but up to 32767).

Default: LINES 66

MARGINB n

Indicates the number of lines allowed for a margin at the bottom of the page.

Default: MARGINB 3

MARGINL n

Indicates the number of spaces allowed for a left hand margin on the report.

Default: MARGINL 0

MARGINT n

Indicates the number of lines allowed for a margin at the top of the page.

Default: MARGINT 3

NEGATIVE L, R, or P

Indicates the printing format of negative numbers:

L	minus sign on left	example -36
R	minus sign on right	example 36-
P	parentheses	example (36)

Default: NEGATIVE L

PAGE (NOPAGE)

Indicates whether you wish INTAC to automatically count lines and skip to a new page on the report whenever the last print line on the page is reached. Report titles and column headings are automatically printed on the top of each new page.

You may override this feature by specifying NOPAGE. Then the test for page breaks occurs only when the TEST-PAGE statement is encountered in the LOGIC section.

Default: PAGE

SEGMENT filename

Assigns a name to the generated report program. INTAC assumes that the report definition file name is defname.DEF, and that the generated BASIC program name is defname.BAS. You may override this assumption by using the SEGMENT option to name the generated program. On RSTS/E operating systems, the name you assign may be any valid RSTS name, including account number, device specification, and protection code. On the VMS operating system the name may be any valid VMS name. An extension of .BAS is assumed if you do not give one on the SEGMENT statement.

The SEGMENT option should be used when the translated program will reside in one account, and the report will be run in one or more other accounts. In this case, include the account number or directory in the SEGMENT filename. The actual generation of the program should occur in the account designated on the SEGMENT statement.

Default: SEGMENT filename

SKELETON filename

Name of skeleton program to be used in generating a report. On the RSTS/E operating system, the standard skeleton INT:INTGES.BAS is normally used. When using the standard skeleton, the report definition file (DEF) must be present in the account when the report is run. The expanded skeleton INT:INTGNS should be specified on RSTS/E systems if the SELECT section of the report definition file contains user BASIC code or more than one INCLUDE/EXCLUDE statement, or if the DEFINE section contains many defined items. The expanded skeleton is the only one available on VMS systems and is the default.

A sophisticated user may modify either INTAC GE skeleton, give the modified version a name, and specify it.

Default: INT:INTGES (RSTS/E system)
INT:INTGNS (VMS system)

WIDTH n

Indicates page width in characters. From this, INTAC determines how many columns will fit on a page.

Default: WIDTH 132

DEFINE SECTION

The DEFINE section allows you to define new data items, perhaps calculated from items in some of the INTAC files accessed for the report. Or these data items may have their values prompted for at run time. These items may be used in titles, headings, or in the LOGIC section like any data item in one of the accessed files. You may also use a DEFINE item to index secondary files, giving you great flexibility in linking files.

In the DEFINE section, you may have up to 99 string fields, up to 99 real numbers, and up to 99 dates and/or integers on VMS systems and on RSTS/E systems if you indicate the expanded skeleton (see the OPTIONS section.) On RSTS/E systems, if you indicate the standard skeleton, you may have up to ten data items of each type. If the DEFINED data item is calculated, the calculation must be defined in the LOGIC section.

DEFINE STATEMENTS

Each line in the DEFINE section defines one data item and has a format similar to that of an INTAC file data item. All items for which the PROMPT option is given must precede those items for which there is no prompt.

The parameters enclosed in brackets are optional. When used, they must occur in the order shown.

FORMAT: itemname type [xx.y] [headings] [PROMPT "prompt msg"]

where

itemname consists of from 1 to 24 characters. The characters must be alphabetic, numeric, or periods (.). No spaces or other special characters allowed. Names which are INTAC or BASIC reserved words may not be used. (See Appendix D for a list of reserved words.)

type data type, as used in INTAC files:
S character string
R real (floating point) number
I integer
D date field

xx.y print format consisting of two elements, xx and y

xx = total width of printed field including commas, decimal points and sign or parentheses. For string fields this is the length of the string. For date fields this is 8 or 10 to indicate the format of the date (default is 8).

y = the number of decimal places in real numbers only.

headings report column headings. If omitted, the itemname will be used as the heading. Enter one to five headings separated from each other by a slash. The entire heading string is enclosed in quotes and separated from the print format by a space.
EXAMPLE: "JANUARY/AMOUNT"

Often, a user wants to include a slash in a column heading. Double slashes must be entered to represent a single slash on the report. Three slashes must be entered if the last character of a heading line is to be a slash.

EXAMPLES:	Definition	Col Heading
	"RATE//SHARE"	RATE/SHARE
	"RATE///SHARE"	RATE/ SHARE

PROMPT "prompt msg"

If this option is given, you will be prompted to enter a value into the item when you run your report. The optional prompt message, if included, will be used as the prompt. Otherwise, the itemname will be used as the prompt.

Prompted items must precede unprompted items in the DEFINE section. When the report is run, prompting is done before SELECT section statements are executed. They are verified for type at the time of entry.

DEFINE SECTION EXAMPLE:

```

200  DEFINE
210  MIN.VALUE I 8 PROMPT "ENTER MINIMUM VALUE TO PRINT"
220  REQUESTOR S 10 PROMPT "ENTER NAME OF RPT. REQUESTOR"
230  END.MO I 3
240  END.YR I 3

```

In this example, the DEFINE item MIN.VALUE will be used in the SELECT section to limit the range of records selected. The item REQUESTOR will be used in a title. END.MO and END.YR are calculated data items that are also included (and the calculations defined) in the LOGIC section.

When the report is run, a dialogue will take place before the SELECT statements are executed:

```

OUTPUT TO <KB:>?
ENTER MINIMUM VALUE TO PRINT? 50
ENTER NAME OF RPT. REQUESTOR? ALEXANDER
26 RECORDS SELECTED

```

SELECT SECTION

The SELECT section has four main functions:

1. To specify up to 3 INTAC files from which data is to be selected
2. To define record selection criteria (rules to INCLUDE and/or EXCLUDE records and the RANGE of records)
3. To define the sort sequence of the selected records
4. To define control break fields on the report. These break fields can then be used in the LOGIC section to specify special processing or headings at breaks.

An INTAC report is prepared from one primary file, called FILE1. This is the main file from which data is to be selected, sorted and printed.

One or two secondary files, FILE2 and FILE3, may also be specified. FILE2 must relate by one of its indexes to fields from FILE1. FILE3 must relate by one its indexes to fields from FILE1 and/or FILE2. Thus, the secondary files are reference files, automatically accessed as each record from the primary file is read. (See Chapter 2.)

You may also insert BASIC code into the SELECT section. See Appendix B for instructions.

SUMMARY OF SELECT STATEMENTS

The SELECT section may contain the statements listed below. The FILE1 statement is required; the others are optional.

FILE1 filename
FILE2 filename INDEXn item1 ... itemn
FILE3 filename INDEXn item1 ... itemn

RANGE INDEXn

INCLUDE IF condition
EXCLUDE IF condition

SORT item1 item2 ... itemn
BREAK item1 item2 ... itemn

HOW INTAC PROCESSES SELECT STATEMENTS

The order of statements in the SELECT section is significant:

RANGE should follow the FILE1 statement, preceding any INCLUDE or EXCLUDE statements for FILE1.

INCLUDE or EXCLUDE statements should be placed after the FILE statement or statements to which they refer or after any RANGE statement for FILE1.

The SORT and BREAK statements should be the last statements in the SELECT section.

INTAC reads data from the files in the following manner:

1. Prompting for RANGE occurs.
2. Each record from FILE1 is read. RANGE checking is performed. If the record is outside the range, the next record is read. If the record is within the specified range, then
3. FILE1 INCLUDE or EXCLUDE tests are made, and if the record is to be excluded, no reading is done from FILE2 or FILE3.

If the FILE1 record is to be included, then the corresponding record on FILE2 is read.

4. INCLUDE or EXCLUDE tests following FILE2 are executed, and if the record should be excluded, no reading is done from FILE3.

FILE3 is read and handled in the same manner as FILE2.

5. If all INCLUDE tests have been passed, the record is selected.

The following examples show how placement of the INCLUDE and EXCLUDE statements can dramatically affect the run time of your report.

EXAMPLES:

```
FILE1 ASSET
FILE2 DEPT INDEX1 DEPT.NO
FILE3 VENDOR INDEX1 VENDOR.NO
    INCLUDE IF PUR.YEAR <72
```

In this first example, each time FILE1 is read, FILE2 and FILE3 are also read, even if the record is not to be selected. If you were including here 20 records out of 2000, you will have done up to 3960 unnecessary reads on the other 2 files.

```
FILE1 ASSET
    INCLUDE IF PUR.YEAR <72
FILE2 DEPT INDEX1 DEPT.NO
FILE3 VENDOR INDEX1 VENDOR.NO
```

This second example shows a more efficient placement of the INCLUDE statement. Since the INCLUDE test is based on FILE1 only, FILE2 and FILE3 will not be read unless the FILE1 record is to be included.

SELECT STATEMENT DESCRIPTIONS

These statement descriptions are in alphabetical order for reference.

BREAK item1 item2 ... itemn[:n1,n2]

The BREAK statement is used to define the control break fields on the report. The break fields are listed from major to minor break field. The first item listed will control BREAK1, the second BREAK2, and so on. You can name a maximum of 8 control break fields. The items can be any data items from FILE1, FILE2, or FILE3. BREAK0 is automatically defined as the highest level break on the report.

You can specify a part of an item to be a break field. This is done by following the item with :n1,n2 where n1 is the starting position of the break field within the item, and n2 is the length of the break field. Partial fields should only be specified for string data items.

The BREAK statement should be placed following all FILE, INCLUDE, EXCLUDE statements, and the SORT statement. Be sure that your SORT statement has the same items as your BREAK statement plus any additional items you wish to sort on.

EXAMPLES:

```
SORT DIVISION DEPT.NO  
BREAK DIVISION
```

```
BREAK ASSET.NO:3;1
```

(Break on the third character in the string; note that ASSET.NO was created as a string type so that partial string sorting and breaking could be done.)

EXCLUDE IF condition

Please see INCLUDE IF condition.

FILE1 filename

This statement specifies the primary file for an INTAC report.

where

filename is the name of an INTAC file (An extension of .INT is assumed if none is given.)

FILE2 filename INDEXn item1 ... itemn[/NOMESSAGE]

This statement specifies an optional secondary file for an INTAC report.

where

filename	is the name of an INTAC file. An extension of .INT is assumed if none is given.
INDEXn	is used to specify the index number from FILE2 that will be used to access the file.
item1...itemn	(separated by spaces) are names of data items from FILE1 or items that are prompted for in the DEFINE section. They are used to create a lookup key for INDEXn of FILE2.
/NOMESSAGE	When the report is run, INTAC will print a warning message if the corresponding record in FILE2 is missing. The /NOMESSAGE OPTION can be used to suppress the printing of this message.

CAUTION: If the corresponding record in FILE2 is missing, reference in SELECT, LOGIC, or CODE sections to items from FILE2 may cause unpredictable results.

EXAMPLE:

```
FILE1 ASSET
FILE2 DEPT INDEX1 DEPT.NO
```

FILE3 filename INDEXn item1 ... itemn[/NOMESSAGE]

This statement specifies another optional reference file. Its format is the same as the FILE2 statement. INDEXn should identify an index from FILE3. The items may refer to items in FILE1 or FILE2 or the DEFINE section. If a name is duplicated on more than one file, be sure to use qualified names.

EXAMPLE:

```
FILE1 ASSET
FILE2 DEPT INDEX1 DEPT.NO
FILE3 VENDOR INDEX1 F1.VENDOR.NO      (Use FILE1 field)
```

INCLUDE IF condition
EXCLUDE IF condition

The INCLUDE and EXCLUDE statements are used to limit the records selected from the files. Only one INCLUDE or EXCLUDE statement may occur after each FILE statement in the SELECT section if the standard skeleton is used on RSTS/E systems. If more than one INCLUDE and/or EXCLUDE statement is used, a record must pass all tests to be selected.

The condition consists of 2 values separated by a logical operator. The values may be INTAC data items, REPORT data items from the DEFINE section, numbers, character strings or dates in quotes, or any valid BASIC variable.

The operators are:

- < LESS THAN
- > GREATER THAN
- = EQUALS
- <= LESS THAN OR EQUAL TO
- >= GREATER THAN OR EQUAL TO
- <> NOT EQUAL
- >> CONTAINS (STRINGS ONLY)

Conditions may be joined by AND or OR. Parentheses can be used to form more complicated conditions.

INCLUDE and EXCLUDE statements should be placed immediately following the FILE statement or statements to which they refer or after any RANGE statement following FILE1.

EXAMPLE:

```
SELECT
  FILE1 ASSET
    INCLUDE IF ASSET.CODE = "P" AND PUR.YEAR >=71
    EXCLUDE IF ORIG.COST <F4.MN.VALUE
  FILE2 DEPT
    EXCLUDE IF F2.DEPT.NO = 8001
```

In this example, only records with an ASSET.CODE of P, purchased during or after the year 1971 (PUR.YEAR is an integer data item) will be selected. In addition, a record will not be selected if the ORIG.COST is less than the DEFINE data item MN.VALUE (which receives its value through prompting). No record with a DEPT.NO of 8001 will be selected. A record must satisfy all of the INCLUDE and EXCLUDE tests to be selected.

DATE EXAMPLE:

```
SELECT
  FILE1 ORDER
    EXCLUDE IF ORD.DATE <"01/15/71"
                                (ORD.DATE is data type D)
```

RANGE INDEXn

This command will significantly increase the efficiency of selecting records, particularly when you are selecting records from a small subset of FILE1. INDEXn is the index number to be used in searching FILE1. The RANGE statement follows the FILE1 statement in the SELECT section. At run time, the user is prompted for starting and ending values of the index. INCLUDE or EXCLUDE tests will be applied only to those records within the specified range.

EXAMPLE:

```
FILE1 ASSET  
  RANGE INDEX1
```

You will be prompted for the starting and ending values of the first index on the ASSET file for records to be selected.

SORT item1 ... itemn[:n1,n2]

The SORT statement specifies the sequence in which to sort the selected records. Enter a list of data items (separated by spaces) from FILE1, FILE2, or FILE3 that are to be used to sort the file. The items are listed from major to minor sort fields. Normally, records will be sorted in ascending order from low to high. However, a descending sort for any item(s) may be obtained by preceding the item(s) with a minus sign.

You may specify a part of an item to be a sort field. This is done by following the item with :n1,n2 where n1 is the starting position of the sort field within the item, and n2 is the length of the sort field. Partial fields should only be specified for string data items.

If the SORT statement is omitted, the report will use INDEX0 of FILE1, the logical record number, as the sequence of the report.

The SORT statement should be placed after all FILE, INCLUDE, and EXCLUDE statements and before the BREAK statement if one exists.

EXAMPLES:

```
SORT DEPT.NO ASSET.NO  
SORT DEPT.NO -ASSET.NO  
SORT ASSET.NO:3,2
```

*(sort low to high on both items)
(sort high to low on ASSET.NO)
(sort on third and fourth characters in string; ASSET.NO was created as a string field so that partial sorting could be done)*

TITLE SECTION

The TITLE section defines the headings that print at the top of each page of the report. Every report program generator definition file should have a TITLE section.

FORMAT: region item1 [...itemn] [+]

where

region indicates the position on the page where the title should print.

LEFT prints on left side of page.

RIGHT prints on right side of page.

CENTER centers the title on the page.

POSn prints the title starting in print position n on the page.

item1 describes the print items. These items may be any of the following

...itemn

DATE prints the current date in the form, DATE: 31-OCT-80.

TIME prints the current time in the form, TIME: 08:41 AM.

PAGE prints page numbers on the report. Page numbers are automatically incremented as each new page is printed.

itemname prints the data item named. May be an INTAC data item or a REPORT data item from the DEFINE section.

"text message" can be any message in quotation marks

Multiple items may be printed on the same print line by listing them separated by spaces.

+ This plus sign is optional; indicates that the title line is continued on the next line. (Note that spaces must precede and follow this symbol.)

STATEMENT SEQUENCE

The TITLE statement begins the section and appears alone on a line. Each subsequent statement begins with the region. Each region statement indicates a new line of heading information, unless the previous line ended with a "+" in which case the heading line is composed of concatenated region statements. Only one region can be specified on a line.

EXAMPLE:

```
400 TITLE
410 LEFT DATE
420 CENT "TRI-CITY COMPANY"
430 LEFT " "
440 CENT "DEPARTMENT - " DEPT.NO " " DEPT.NAME
450 CENT "MONTHLY DEPRECIATION ON ALL PURCHASED ITEMS"
```

460 CENT "STRAIGHT LINE METHOD"
470 LEFT " " ! SKIPS A LINE

DATE: 03-Jul-82

TRI-CITY COMPANY

DEPARTMENT - 6501 PLANNING
MONTHLY DEPRECIATION ON ALL PURCHASED ITEMS
STRAIGHT LINE METHOD

COLUMNS SECTION

The COLUMNS section is used to define columns of data on the report, the format of the data in the columns, and the column headings. The COLUMNS section contains one line for each column on the report. Each line may be associated with a previously defined data item from the INTAC files used in the report or from the DEFINE section. The definition of the data item includes a print format that will be used as the default width of the column unless you override that width in this section.

You have additional control over the placement of columns across the width of the report if you specify a character position number. Usually, you can estimate the correct position, making adjustments after running a report. A useful tool is a formatting ruler that measures 150 print positions.

FORMAT: [C] [position1] [itemname] [headings] [xx.y]
 [L] [position1] [position2]
 [R]

where

C, L, R	indicates center, left, or right justification of the data within the column. Default is R for numbers, L for character strings. For numbers, only R is valid.
position1	is a number to indicate the beginning position of the column on the page. position1 defaults to 2 positions past the end of the prior column.
position2	is a number indicating the ending position. position2 defaults to position1 plus the width of the column minus 1.
itemname	is any INTAC or DEFINE data item name. Either an itemname or heading is required.
headings	are optional report column headings (a heading must be specified if no itemname is given); may be used to override the headings associated with the itemname. Enter one to five headings separated from each other by a slash. (If the underline option described below is used, the maximum number of headings is four.) The entire heading string is enclosed in quotes.

Example: "TOTAL/PURCHASES"

Often, a user wants to include a slash in a column heading. Double slashes must be entered to represent a single slash on the report. Three

slashes must be entered if the last character of a heading line is to be a slash.

EXAMPLES:	Definition	Col Heading
	"RATE//SHARE"	RATE/SHARE
	"RATE///SHARE"	RATE/ SHARE

xx.y is the optional print format used to override the print format associated with itemname.

xx = total width of printed field including commas, decimal points, sign or parentheses.

y = the number of decimal places in floating point numbers only.

GLOBAL OPTION

Options that apply to all of the columns in a report are specified in the COLUMNS statement. The following global option is available.

Uc underline each column heading with the character c (any printable character). U- would underline the headings with dashes.

EXAMPLE OF COLUMN WITH POSITION OPTION

510 L 40 ASSET.NO

(This column will appear at position 40 with data left justified within the column)

EXAMPLE OF A COMPLETE COLUMNS SECTION

```

500 COLUMNS U-
510 ASSET.NO
520 DESCRIPTION
530 ASSET.TYPE
540 ORIG.COST
550 LIFE
560 PUR.MONTH "BEG/MO"
570 PUR.YEAR "DTE/YR"
580 "END/MO" 3
590 "DATE/YR" 4
600 "MONTHLY/DEPR"

```

In conjunction with the DISPLAY statements in the LOGIC section, the COLUMN section above causes the following columns to appear on the report:

ASSET NO	DESCRIPTION	ASSET TYPE	ORIG.COST	LIFE	BEG MO	DTE YR	END MO	DATE YR	MONTHLY DEPR
00008	PROJECTOR	E	256.10	5	10	71	9	76	4.27

LOGIC SECTION

The LOGIC section specifies the calculations to be performed during the printing of a report. In conjunction with the COLUMN section, it may be used to specify the data to print and optionally, the format of the data. The LOGIC section allows the full flexibility of BASIC statements as well as INTAC statements. If you are an experienced programmer, see Appendix B for additional information about writing sophisticated GE programs. INTAC also has many built-in functions that may be used in the LOGIC section. See Appendix C for their use.

The LOGIC section is divided into subsections describing the calculations to be performed for each detail record, and those to occur at the beginning and the end of control breaks.

LOGIC SUBSECTIONS

The subsections of the LOGIC section are named as follows:

```
BEGIN REPORT
BEGIN BREAK itemname
              breaknumber

DETAIL PROCESSING
END BREAK itemname
              breaknumber

END REPORT
```

The format of the BEGIN and END statements is:

```
BEGIN [BREAK] itemname
              breaknumber

END    [BREAK] itemname
              breaknumber
```

The word BREAK is optional. The itemname must be one of the itemnames on the BREAK statement in the SELECT section. If you use breaknumber, it must be a break defined on the BREAK statement. The first item listed on the BREAK statement defines breaknumber1, etc. (The itemnumber may not be used here.) If your BREAK statement uses a partial field, you must use the breaknumber rather than the itemname on your BEGIN and END break statements.

The following are valid BEGIN and END subsection names:

```
BEGIN BREAK 3
BEGIN 3
BEGIN DEPT.NO
BEGIN BREAK DEPT.NO
END BREAK 3
END 3
END DEPT.NO
END BREAK DEPT.NO
```

There may be up to 9 BEGIN and 9 END subsections, including BEGIN REPORT, END REPORT, and one for each control break defined with the BREAK statement in the SELECT section.

The format of the DETAIL statement is DETAIL [PROCESSING]

The DETAIL subsection is required. LOGIC statements contained in the DETAIL subsection are performed once for each record processed.

All LOGIC statements must follow one of the subsection headers. The following example shows the SELECT section and the LOGIC sections that could be used for a depreciation report. (If you do not want any special headings or processing at a particular break, you do not need to include BEGIN or END statements for it.)

```

300  SELECT
310      FILE1 ASSET
320      INCLUDE IF ASSET.CODE = "P"
330      FILE2 DEPT INDEX1 DEPT.NO
340      SORT DEPT.NO PUR.YEAR PUR.MONTH ASSET.NO
350      BREAK DEPT.NO PUR.YEAR

3500  LOGIC
5000      BEGIN REPORT                                !BREAK0
6000      BEGIN DEPT.NO                                !BREAK1
6010      NEW-PAGE
12000     DETAIL PROCESSING
12010         MONTHLY.DEPR = ORIG.COST/(LIFE * 12)
12020         END.YR = PUR.YEAR + LIFE
12030         END.MO = PUR.MONTH - 1
12040         IF END.MO = 0 THEN &
                     END.MO = 12 &
                     END.YR = END.YR - 1
12050         DISPLAY ASSET.NO:COL1 DESCRIPTION ASSET.TYPE ORIG.COST LIFE +
12060         DISPLAY PUR.MONTH:COL6 PUR.YEAR END.MO END.YR MONTHLY.DEPR
12070         TOTAL ORIG.COST
13000     END PUR.YEAR                                !BREAK2
13010         DISPLAY "-----":COL4
13020         DISPLAY "TOTAL PURCHASES FOR YEAR ":5 PUR.YEAR:30 T2.ORIG.COST:COL4
13030         SKIP1
14000     END DEPT.NO ~                                !BREAK1
14010         DISPLAY "-----":COL4
14020         DISPLAY "TOTAL PURCHASES .....":5 T1.ORIG.COST:COL4
15000     END REPORT                                !BREAK0

```

GE

LOGIC STATEMENTS SUMMARY

The following special INTAC statements may be included in the LOGIC section. You may also include any valid BASIC statement. INTAC has many built-in functions that may also be used in the LOGIC section. Appendixes B and C give technical information for using BASIC code and INTAC functions.

DISPLAY	prints a report line.
HEADINGS	turns printing of column headings at page breaks ON or OFF as requested. Default is ON.
NEW-PAGE	skips to a new page on the report.

SKIPn	prints n blank lines on the report.
TEST-PAGE	checks for page overflow and prints new headings if appropriate.
TOTAL	computes totals for designated items.

LOGIC STATEMENT DESCRIPTIONS

DISPLAY element1 element2 ... elementn [+]

The DISPLAY statement is used to specify a series of data elements to print. It may also be used to specify a print position and format. It is used in conjunction with the COLUMNS section which has defined a series of columns and the format of a print line. You may control the placement of data elements by specifying column position or character position. It is usually possible to estimate an appropriate character position; however, you may also use a formatting ruler to keep track of character positions.

Each element in the DISPLAY statement may take the form

itemname:[position][,format]

where

itemname	is any INTAC data item name, DEFINE data item name, or BASIC variable. It may also be text enclosed in quotes; for example, to provide a label for a total at BREAK1 DISPLAY "TOTAL HOURS":1 T1.HOURS
position	may be indicated either by COLn where n is the column as defined in the COLUMNS section or by n, a specific print position on the page. In both of these options, n may be a number or an expression that can be evaluated at execution time. If a position is omitted, the item will print in the next available column defined in the COLUMNS section.
,format	takes the form ,xx.y xx.y is an INTAC print format. If it is omitted, the format is taken from the COLUMNS section or item definition. xx is the width of printed field including commas, decimal points, sign, or parentheses y is the number of decimal places in floating point numbers only.

If a DISPLAY statement ends with a plus sign (+), the print line is to be continued on another DISPLAY statement on the next line. The + sign must be preceded by a space. The second display statement must specify position for the first item being displayed.

If DISPLAY occurs alone as a statement, the line defined in the COLUMNS section is printed.

EXAMPLE 1: 6030 DISPLAY ASSET.NO DESCRIPTION ORIG.COST END.YEAR

In this example, each of the data items will be printed in the columns defined in the COLUMNS section of the report definition according to the specifications used in the COLUMNS section (position and format). All of the elements except for END.YEAR are data items in the INTAC file and the values from the file will be printed. END.YEAR was defined as a report variable, and the calculated results will print.

EXAMPLE 2:

```
6020 DISPLAY ASSET.NO:COL1 DESCRIPTION:10 ORIG.COST:40,9.0 +
6030 DISPLAY LIFE:55
```

In the above example, the first data element (ASSET.NO) is to be printed in COL1 as defined in the COLUMNS section. Since ASSET.NO has a print format of 5, this column will take up the first 5 positions across the page. The second element (DESCRIPTION) will be printed starting at position 10 on the page using its INTAC file default print format which is 22. The third element, (ORIG.COST) will be printed in position 40 on the page, using a format of 9.0. The field defined in the second DISPLAY statement (LIFE) will print on the same line in print position 55. Note that LIFE must have a position specified because it is the first item in a continued DISPLAY statement.

EXAMPLE 3: 6480 DISPLAY "-----":COL4

The underline will appear under the fourth column, which is ORIG.COST.

EXAMPLE 4: 7300 DISPLAY

In this example, the print line generated contains all items defined in the COLUMNS section.

HEADINGS ON/OFF

The HEADINGS statement can be used to override the automatic printing of column headings at page breaks. The default is HEADINGS ON.

NEW-PAGE

Whenever the statement NEW-PAGE is encountered in the LOGIC section, INTAC will skip to a new page on the report and print the titles and column headings. INTAC automatically performs the NEW-PAGE function each time it reaches the bottom of a page. You need to specify NEW-PAGE only if you would like additional page breaks to occur, such as at the beginning or ending of a specific control break.

In the example, the NEW-PAGE command results in each new DEPT.NO starting on a new page.

EXAMPLE:

```
6000 BEGIN DEPT.NO
6010      NEW-PAGE
```


SKIPn

The SKIPn statement causes n blank lines to print on the report.

Suppose you wanted the detail lines on the report to be double spaced.

EXAMPLE:

```
3500 LOGIC

12000 DETAIL PROCESSING
12010     TOTAL ORIG.COST
12020     DISPLAY
12030     SKIP1
```

Note that SKIPn does not generate blank lines if at the end of a page. Use DISPLAY " " to achieve such blank lines.

TEST-PAGE

Use the TEST-PAGE statement if you wish to override the automatic line counting and page break feature of INTAC.

If you have used the PAGE feature in the OPTIONS section of the GE file definition, INTAC automatically skips to a new page and prints page headings whenever the line counter reaches the last print line on the page. The last print line number is determined from the LINES and MARGINB options in the OPTIONS section. This automatic test for end of page occurs before each DISPLAY statement in the report definition.

If you wish to override this feature, then specify NOPAGE in the OPTIONS section. Then, the test for page overflow will occur in the LOGIC, only when the TEST-PAGE statement is encountered.

TOTAL

```
itemname1 ... itemnamen
```

The TOTAL statement generates totals at each control break for all data items specified in the statement. The data items named may be INTAC file data items or REPORT items from the DEFINE section. There may be any number of TOTAL statements in the LOGIC section (up to 40 items can be totaled).

Totals can be generated at each control break defined on the BREAK statement. The totals are accumulated in data items (or, more technically, in variables) named Tn.itemname, where n is the break level number. These totals may be thought of as automatically defined data items. You may use these totals in the LOGIC section as though they were data items in the file. You will probably want to DISPLAY the totals, using a heading enclosed in quotations. For example,

```
12030 DISPLAY "TOTAL PURCHASES FOR YEAR":5 PUR.YEAR:30 T2:ORIG.COST:COL4
```

EXAMPLE:

```
300  SELECT
      .
360      BREAK DEPT.NO PUR.YEAR
      .
3500  LOGIC
12000      DETAIL PROCESSING
12010      TOTAL ORIG.COST
12020      DISPLAY
```

In this example, 2 control break fields are defined. The following chart summarizes the names used to accumulate totals for the TOTAL statement.

<u>BREAK LEVEL</u>	<u>BREAK FIELD</u>	<u>TOTAL FIELD NAMES</u>
1	DEPT.NO	T1.ORIG.COST
2	PUR.YEAR	T2.ORIG.COST

These items containing the accumulated totals are available to you in the LOGIC section. The totals for each control break level are automatically zeroed out at the end of the corresponding END BREAK subsection.

The following example shows computing and printing control totals:

EXAMPLE:

```
4090      DISPLAY ASSET.NO:COL1 DESCRIPTION ASSET.TYPE ORIG.COST LIFE +
4100      DISPLAY PUR.MONTH:COL6 PUR.YEAR END.MO END.YR MONTHLY.DEPR
4110      TOTAL ORIG.COST
4120  END PUR.YEAR
4130      DISPLAY "-----":COL4
4140      DISPLAY "TOTAL PURCHASES FOR YEAR ":5 PUR.YEAR:30 T2.ORIG.COST:COL4
4150      SKIP1
4160  END DEPT.NO
4170      DISPLAY "-----":COL4
4180      DISPLAY "TOTAL PURCHASES .....":5 T1.ORIG.COST:COL4
4190  END REPORT
```

GE

CODE SECTION

The CODE section is an optional section reserved for user subroutines and functions. The experienced user will be able to take full advantage of the flexibility of INTAC by using the CODE section. Additional technical information is offered in Appendixes B and C.

If you are relatively inexperienced with BASIC programming, this part of the manual will give you a sense of what can be achieved so that you may ask for assistance when your report is very complex.

A subroutine is a programming segment that performs a distinct set of procedures. Usually, a subroutine is self-contained enough to be thought of as independent of the rest of the LOGIC section. Place such independent procedures in the separate CODE section rather than in the LOGIC section in order to avoid duplicating instructions.

A function is usually a calculation standard in business applications. There are also many functions used to access INTAC files. A function returns a value for use like a data item in your report. INTAC has many built-in functions that you may reference in either the LOGIC or CODE section. (See Appendix C for functions.)

USE OF THE CODE SECTION

In the report definition file, lines 20000 through 23999 are reserved for the CODE section. The line numbers specified in the CODE section are not changed by the INTAC report generator as they are in other sections. Thus you have control of the line numbers and this control allows you to use references to specific CODE lines in the LOGIC section. Such references are called GOSUBS for the BASIC command GOSUB, or "go to the subroutine."

Organize the LOGIC and CODE sections to take advantage of subroutines to avoid duplication of instructions that might be needed in several places in the section. The same subroutine can be called many times in the LOGIC section.

The CODE section may contain any BASIC or LOGIC section statements and variables. If the CODE section is used, it must be the last section in the Report definition file.

EXAMPLES:

The following example shows the Tri-City depreciation report definition with calculations for monthly depreciation taken out of the LOGIC section and placed in the CODE section:

```

3500 LOGIC
4010     BEGIN REPORT
4020     BEGIN DEPT.NO
4030         NEW-PAGE
4040     BEGIN PUR.YEAR
4050     DETAIL PROCESSING
4060         GOSUB 20100
4070         DISPLAY ASSET.NO:COL1 DESCRIPTION ASSET.TYPE ORIG.COST LIFE +
4080         DISPLAY PUR.MONTH:COL6 PUR.YEAR END.MO END.YR MONTHLY.DEPR
4090         TOTAL ORIG.COST
4100     END PUR.YEAR
4110     DISPLAY "-----":COL4
4120     DISPLAY "TOTAL PURCHASES FOR YEAR ":5 PUR.YEAR:30 T2.ORIG.COST:COL4
4130     SKIP1
4140     END DEPT.NO
4150     DISPLAY "-----":COL4
4160     DISPLAY "TOTAL PURCHASES .....":5 T1.ORIG.COST:COL4
4170     END REPORT
20000 CODE
20100     MONTHLY.DEPR = ORIG.COST/(LIFE * 12)
20110     END.YR = PUR.YEAR + LIFE
20120     END.MO = PUR.MONTH - 1
20130     IF END.MO = 0 THEN &
           END.MO = 12 &

```

```

20140      \END.YR = END.YR - 1
          RETURN

```

This line returns the report generator to line 4070

The second example below is not based upon the Tri-City case study. The user has created her own BASIC function for the CODE section and called it repeatedly in the LOGIC section to do customized printing. She thereby avoids repeating this code in the LOGIC section. (INTAC functions are included in the LOGIC and CODE sections in the same way that this user-defined function has been included. See Appendix C.)

```

3500 LOGIC

.
.
.
4500 DETAIL PROCESSING
4510     COMPANY=LEFT(CD.NAME,20%)
4520 END PROG.NO
4530     DISPLAY PROJ.NO PROG.NO COMPANY SETUP +
4540     MY.ARRY(Y.X%)=MY.DATA(Y.X%) FOR Y.X%=1% TO 12%
4550     Y.X%=FNX.PRINT%                ! Call the function
4560 END PROJ.NO
4570 END SALESPERSON
4580     DISPLAY "TOTAL FOR SALESPERSON" +
4590     MY.ARRY(Y.X%)=MY.DATA(Y.X%) FOR Y.X%=1% TO 12%
4600     Y.X%=FNX.PRINT%                ! Call the function
4610 END REPORT
4620     DISPLAY "BRANCH TOTAL" +
4630     MY.ARRY(Y.X%)=MY.DATA(Y.X%) FOR Y.X%=1% TO 12%
4640     Y.X%=FNX.PRINT%                ! Call the function
20000 CODE
20010 ! *****
!             MY OWN PRINT FUNCTION&
! *****

20110     DEF FNX.PRINT%                ! BASIC code statements
20120     FOR Z.Z%=5% TO HCT%
20130         IF MY.ARRY(Z.Z%-4%)<>0 THEN&
             DISPLAY MY.ARRY(Z>Z%-4%):COLZ.Z%,10.2
20140     NEXT Z.Z%&

20160     Z.TOT=0 &
\Z.TOT=Z.TOT+MY.ARRY(Z.Z%) FOR Z.Z%=1% TO MAX.MO% &

20170     Z.Z%=HCT%+1%
20180     DISPLAY Z.TOT:COLZ.Z%,10.2
20190     FNEND&

```

REPORT PROGRAM GENERATION

Once you have created a report definition file, you must translate the definition into a BASIC program by using the GENERATE facility. Then you may run the program whenever you need to produce the report. The program is run at the system level.

THE GENERATE FACILITY

The INTAC GENERATE facility is used to generate a BASIC report program from an INTAC report definition file. The following example shows a complete GENERATE session and a system command to run the report. Notice that you enter the facility by typing the GE command at the COMMAND? prompt.

COMMAND? **GE**

GENERATE AN INTAC REPORT

REPORT DEFINITION FILE? **DEPR.DEF**

Enter the name of your report definition file, followed by the options selected. An extension of .DEF is assumed if none is given. INTAC begins processing the report file definition.

GENERATE AN INTAC REPORT, PART 2

184 LINES, FILE: USR:DEPR.BAS
BEGINNING COMPILATION OF SEGMENT: USR:DEPR(BP)

COMPILE CONTINUING
END OF COMPILE GENERATED PROGRAM
COMMAND?

Leave INTAC with a carriage return.

END OF INTAC

Ready

RUN DEPR

Run the report at the system level.

REPORT DEPR

OUTPUT TO <KB: >? **DEPR/FORM**

See page 10-39 for options for running the report.

42 RECORDS SELECTED

END OF REPORT DEPR

Occasionally, the GE process will display a warning message (preceded by a percent sign) generated by the BASIC compiler. For example, a common message is "% inconsistent function usage at lineno." These warning messages do not interfere with the GE process and may be ignored. If a serious problem exists in the GE, an error message will be displayed and the process will not proceed.

GENERATE FACILITY OPTIONS

Any logical combination of options may be specified. Options are separated from the filename and from each other by a slash. If no options are specified the default is to compile the program in BASIC and not to run the program.

The available options are:

/BP2 on the RSTS/E system, compile in BASIC-PLUS-2 (default is BASIC-PLUS)

/DIS	display the commands used to compile the program
/KB:	output generated code to terminal (should use /NOS too—for debugging purposes)
/NCO	don't compile
/NOK	don't delete source code; reportname.BAS (created by GE) is saved for debugging purposes
/NOS	don't append skeleton code (for debugging purposes)
/RUN	run the program after generation

EXAMPLE:

```
COMMAND? GE
GENERATE AN INTAC REPORT
REPORT DEFINITION FILE? DEPR.DEF/KB:/NOS
```

This command will result in a display of the generated code at the terminal, without the skeleton code.

ERROR MESSAGES:

A very common error message informs you that a heading is too wide for the print format of the item. Your report file definition will be processed anyway; the width of the column will be extended to be the width of the heading. All subsequent columns will be adjusted.

```
WARNING: COLUMN WIDTH AT 230 LESS THAN HEADING, WIDTH INCREASED
```

SUBSEQUENT REPORTS

Once the BASIC program has been generated from the report definition file you may produce the report by entering the command:

```
RUN reportname
```

(See below, "Obtaining Reports.")

OBTAINING REPORTS

Once you have created the report definition file and used the GENERATE facility to translate it into a report program, you may obtain the report by simply running the program at the system level, using the appropriate system commands. (On both the RSTS/E and the VMS systems, the command is RUN filename; however, different print specifications are available on these two different computer systems).

Once you have given the system command to run the program, the following question is asked.

OUTPUT TO <KB:>?

If you want to see the report at the terminal, simply press the Return key.

Below are procedures for printing or saving reports.

1. To Print on ROSS SYSTEMS Line Printers

Enter LP: if you wish the report to be printed on the line printer, or LPn: to use an alternative line printer, where n is the number of the line printer. On the RSTS/E systems, these device names may be followed by any of the RSTS/E system QUE specifications for number of copies, etc. See the *Ross Systems 11/70 User's Guide* for details.

On the VMS system, these device names may be followed by any of the VMS PRINT specifications.

2. To Save the Report in a File

Enter the name of a file if you wish the report to be printed to a file. If no extension is given, an extension of .RPT is assumed, and any existing file with that name is written over.

Three additional options may be entered when saving the report in a file in response to the question OUTPUT TO <KB:>?

/FORM or /NOFORM
/APPEND
/REPLACE

Each of these options may be abbreviated to the first three letters.

The use of FORM or NOFORM dynamically at run time will override the specification in the OPTIONS section of the report definition file.

The APPEND option allows you to append generated reports to existing reports of the same name.

The REPLACE option replaces an existing file with the same name.

EXAMPLES:

```
OUTPUT TO <KB:>? DEPR/FORM
OUTPUT TO <KB:>? DEPR/NOFORM
OUTPUT TO <KB:>? DEPR/APPEND/NOFORM
OUTPUT TO <KB:>? LP:/NOF
OUTPUT TO <KB:>? /NOFORM
OUTPUT TO <KB:>? /FOR
```

Here is a terminal session to run a report, sending it to a file called DEPR and using the FORM option.

Ready

RUN DEPR

REPORT DEPR

```
OUTPUT TO <KB:>? DEPR/FORM
42 RECORDS SELECTED
```

END OF REPORT DEPR

GE

GENERATE PLANNING WORKSHEET

```

11  !*****
12  !GENERATE FILE DEFINITION NAME:
13  !PURPOSE OF FILE:
14  !AUTHOR:
15  !DATE CREATED:
16  !*****
18  !
100 OPTIONS

```

COMMA/NOCOMMA	MARGIN	SEGMENT
FORM/NOFORM	NEGATIVE L/R/F	CHAIN
LINES	WIDTH	SKELETON
MARGINB	PAGE/NOPAGE	
MARGINL	MOVE/NOMOVE	

>200 DEFINE

FORMAT: itemname type [xx.y] [headings] [PROMPT "prompt msg"]

300 SELECT

The SELECT section may contain the statements listed below. The FILE1 statement is required; the others are optional.

```

FILE1 filename
FILE2 filename INDEXn item1 ... itemn
FILE3 filename INDEXn item1 ... itemn
RANGE INDEXn          (may vary in placement)
INCLUDE IF condition  (place after relevant statement)
FILE EXCLUDE IF condition
SORT item1 item2 ... itemn
BREAK item1 item2 ... itemn
                        (place SORT and BREAK last in section)

```

400 TITLE

```

FORMAT: region item1 [...itemn] [+]
LEFT DATE +
RIGHT PAGE
LEFT TIME +
CENT

```

500 COLUMNS U-

FORMAT:

```
[C] [position1] [position2] [itemname] [headings] [xx.y]
[L] [position1] [position2] [itemname] [headings] [xx.y]
[R] [position1] [position2] [itemname] [headings] [xx.y]
```

3500 LOGIC

```
BEGIN REPORT                !BREAK0
BEGIN item1                  !BREAK1
BEGIN item2                  !BREAK2
DETAIL PROCESSING
END item2                    !BREAK2
END item1                    !BREAK1
END REPORT                   !BREAK0
```

20000 CODE

```
20100 !*****
20110 !*** SUBROUTINE
20120 !*****
```

Common LOGIC and CODE Statements (BASIC Statements also available)

DISPLAY element1,...,elementn [+]

TOTAL item1 item2 ... itemn

NEW-PAGE

SKIPn

TEST-PAGE

HEADINGS ON/OFF

GE