

GC21-7514-2
File No. S3-28

Systems

**Introduction to RPG II
Report Writing**

IBM



GC21-7514-2
File No. S3-28

Systems

**Introduction to RPG II
Report Writing**

IBM

Third Edition (July 1978)

This is a major revision of, and obsoletes, GC21-7514-1. Major changes include using the latest level specification forms and updating the terminology.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, be sure you have the latest edition and any technical newsletters.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. Use this publication only for the purpose stated in the *Preface*.

Publications are not stocked at the address below. Requests for copies of IBM publications and technical information about the systems should be made to your local IBM representative or to the branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. Use the Reader's Comment Form at the back of this publication to make comments about this publication. If the form has been removed, address your comments to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the the information you supply.

This book is intended for persons who want to learn to write programs which will produce printed reports using the RPG II language. The book is designed to be used by a person with no previous knowledge of computers and programming, or by a person who already knows a programming language but wants to learn about report writing in RPG II.

After reading this book, you should not expect to be able to write complex RPG II programs. You will be introduced to only a small part of RPG II. Many important features in RPG II are not discussed in this manual. The RPG II reference manual for your system contains all of the information about RPG II features on your system. You should have gained enough background knowledge so that you can readily learn more detailed information—either from reference manuals, classes, or IBM personnel—which is required for writing programs for your system.

How This Manual Is Organized

The first chapter describes, in general terms, how a system operates and the things you must do to run a particular job. The information presented should answer such questions as:

- What are the parts of a data processing system?
- What is a program?
- What is a programming language?
- What is an RPG II program?
- How is an RPG II program run on a system?

The second chapter describes the RPG II program cycle and the RPG II specifications you must write to do a particular task. The material in this chapter is arranged to provide a gradual development of concepts, proceeding from the simple to the more complex. Thus, it is important to read the material in sequence. Sample jobs are used to illustrate the concepts presented.

The third chapter explains an RPG II programmer's job more fully. It shows, by means of a sample job, the things a programmer must do from the start of a job to its completion.

RPG II Coding Forms

Following is a list of forms used to code and debug RPG II programs. Contact your local IBM branch office to order any of these materials:

- RPG Control and File Description Specifications, GX21-9092
- RPG Input Specifications, GX21-9094
- RPG Calculation Specifications, GX21-9093
- RPG Output Specifications, GX21-9090
- RPG Indicator Summary, GX21-9095
- RPG Debugging Template, GX21-9129
- Printer Spacing Chart, X20-1776
- Record Layout Form, X21-9088
- Disk File Layout Chart, GX21-9108
- Printer/Display Layout, GX21-9174

CHAPTER 1. BASIC DATA PROCESSING AND PROGRAMMING CONCEPTS	1		
Parts of a Data Processing System	1		
Input Devices	1		
Output Devices	1		
Processing Unit	1		
Programs and Programming Languages	3		
Source Programs	3		
Source Programs to Object Programs	5		
Summary	6		
Data Processing Terms and Programming Aids	7		
CHAPTER 2. RPG II PROGRAMMING LANGUAGE	11		
RPG II PROGRAM CYCLE	11		
WRITING SPECIFICATIONS FOR INPUT AND OUTPUT OPERATIONS	14		
Program Cycle Operations	14		
Describing the Files	16		
File Names	17		
Device Designation	18		
File Use	18		
File Designation	18		
Record Size	19		
Describing Input Records	20		
File Names	21		
Field Names	22		
Field Location	24		
Type of Data	24		
Describing Output Records	25		
File Names	26		
Record Type	27		
Field Names	29		
Field Location	29		
Printed Reports	30		
Spacing	30		
Skipping	31		
Editing	33		
Job 1 (TRNREG): Printing a Simple Report Using the Three Basic Cycle Operations	36		
Job Definition	36		
Job Requirements	37		
Job Specifications	38		
WRITING SPECIFICATIONS FOR CALCULATION OPERATIONS	40		
Program Cycle Operations	40		
Describing Type of Operation	42		
Describing Data to be Used	42		
Describing the Result Field	45		
Result Field Length	47		
Decimal Positions	47		
Half-Adjusting Results (Rounding)	47		
Job 2 (TRNREG): Doing Simple Calculations	49		
Job Definition	49		
Job Requirements	50		
Job Specifications	51		
WRITING SPECIFICATIONS FOR INDICATORS	53		
Control Level Indicators	54		
Program Cycle Operations	56		
		RPG II Specifications	57
		Using the Blank-After Specification	59
		Job 3 (TRNREG): Using Control Level Indicators to Calculate and Print Totals	60
		Job Definition	60
		Job Requirements	61
		Job Specifications	62
		First Page Indicator	64
		Program Cycle Operations	65
		RPG II Specifications	66
		Overflow Indicators	68
		Program Cycle Operations	70
		RPG II Specifications	72
		Using Spacing with Overflow	72
		Using Overflow and 1P Indicators Together	74
		Last Record (LR) Indicator	74
		Program Cycle Operations	75
		RPG II Specifications	76
		Job 4 (TRNREG): Using First Page, Overflow, and Last Record Indicators to Print Headings and Totals	77
		Job Definition	77
		Job Requirements	77
		Job Specifications	79
		Record Identifying Indicators	81
		Program Cycle Operations	82
		RPG II Specifications	83
		Specifying Record Identification Codes	85
		Specifying Record Identifying Indicators	86
		Specifying Record Type Sequence	87
		Job 5 (STOKST): Using Record Identifying Indicators to Process Different Record Types	89
		Job Definition	89
		Job Requirements	89
		Job Specifications	92
		Resulting Indicators	96
		Program Cycle Operations	96
		RPG II Specifications	98
		Using the Compare Operation	99
		Using an Arithmetic Operation	102
		Job 6 (STOKST): Using Resulting Indicators to Test Contents of Result Fields	105
		Job Definition	105
		Job Requirements	105
		Job Specifications	108
		Field Indicators	111
		Program Cycle Operations	111
		RPG II Specifications	112
		Job 7 (AGETB): Using Field Indicators to Test Contents of Input Fields	114
		Job Definition	114
		Job Requirements	114
		Job Specifications	116
		Conditioning Operations by More Than One Indicator	118

CHAPTER 3. THE PROGRAMMER'S JOB	121
Determining the Job Requirements	122
Determine RPG II Specifications Needed for the Job	127
Write the Specifications	129
Document the Program	133
Prepare for Compilation	134
Specifications Form Order	134
Control Specifications Preparation	135
Checking the Specifications	135
Compile the Source Program	137
Test the Program	140
GLOSSARY	141
INDEX	145

Chapter 1. Basic Data Processing and Programming Concepts

PARTS OF A DATA PROCESSING SYSTEM

Data processing systems differ widely in appearance, usually consisting of several connected units. Regardless of their size or shape, however, all data processing systems have common features.

Data processing systems operate electronically. They are complex machines with thousands of circuits and wires. Fortunately, you do not have to understand the purpose of each circuit or wire. To write a program which will produce a report, all you need to know are the purposes of the main parts of a data processing system: the input devices, the output devices, and the processing unit.

Input Devices

Data you give a system to work with is called *input*. The device used for getting that data into the system is called an *input device*. Several kinds of input devices may be used; among them are card readers, disks, diskettes, tapes, and keyboards. For the purposes of this manual, specific input devices will not be discussed. Information about the input devices applicable to your system is in the RPG II reference manual for your system.

Output Devices

Data produced by a system is called *output*. The device that produces the output is called an *output device*. Several kinds of output devices may be used; among them are card punches, disks, diskettes, tapes, printers, and display screens. For the purposes of this manual, specific output devices will not be discussed. Information about the output devices applicable to your system is in the RPG II reference manual for your system. Because this manual deals only with report writing in RPG II, we will refer only to printed output.

Processing Unit

The main part of a data processing system is the processing unit. The processing unit can be divided into three sections—storage, control, and arithmetic/logic—according to the special function each performs.

Storage

Storage is the system's memory area. This area is divided into many storage positions which the system uses to electronically store information. The actual number of positions in storage depends upon the size of the storage unit. Each storage position has an identifying number called an *address*.

A storage address serves the same purpose as a house address. Information is sent to and received from these locations. The information can be easily retrieved using the address where the information is stored.

Arithmetic/Logic

Calculations (such as add, subtract, multiply, and divide) are performed in the arithmetic/logic unit. When your instructions tell the system to do an operation such as add, the information to be added is transferred from storage to the arithmetic/logic unit. The operation is then performed and the result is sent back to storage.

Control

The control section is the system's decision maker. It retrieves instructions from storage, determines what has to be done, and directs other units or devices to perform the required operations.

PROGRAMS AND PROGRAMMING LANGUAGES

Data processing systems do only what you tell them. When you give a system instructions, however, it might seem as though the system requires more than you would need to do the same job. But remember, a data processing system cannot think: it requires explicit instructions, even for those things you would do almost without thinking.

When you are to do a job yourself, you need three basic things:

- Information to work with (input).
- Instructions telling you how to work with (process) the information.
- Additional instructions describing the expected results (output).

In data processing terms, input is what you put into the system, processing is what the system does with the input, and output is the result of processing. Every job you run on a system has these three parts. You write instructions to describe what you want the system to do with each part. These instructions are called a *program*.

To communicate with the system, you must use the system's language, or one that can be translated into that language. The system's language is called *machine language*. It consists of letters, numbers, and symbols that, when properly arranged, have a specific meaning to the system and, when interpreted by the system, cause it to perform a desired function.

Because machine language is so very different from our own language, it is extremely difficult to use it to write a program. For this reason, programming languages have been created. A programming language allows the programmer to use familiar words and symbols to write instructions.

The RPG II programming language is composed of letters, numbers, and symbols which you put together to form an instruction (express a thought). When creating instructions in the RPG II language, you must follow certain rules just as you would when constructing a sentence in English. You will learn about these rules in the second part of this manual.

The set of instructions you write is called a *source program*. This source program is translated by a program called the compiler, resulting in a machine language program called the *object program*. It is the object program that you use to do a job. In fact, you can use it over and over to do the same job.

Source Programs

The instructions you write for any program must describe the input, processing, and output requirements of the job. For example, one instruction might direct the system to read an input record, another might specify the adding of two numbers, and another may tell the system to print a line on the printer. Because all jobs are not the same, you provide a different set of instructions (program) for each job.

To describe the input, processing, and output requirements of your job, you supply different information on each form. For example, you have to describe what your input data is like and specify the device (such as a disk unit) that will read it. You also have to describe how the input data is to be processed. This includes specifying what type of operations (such as add or subtract) must be performed upon the data. Finally you specify what kind of output you want (such as a printed report), what information must be included in the output, and how that information should be arranged.

After you have coded the specifications forms, the next step is to get the coded information into the system. The system cannot read the coded forms, so you must put the specifications into a format that the system can read. Depending on your system, you key the specifications onto a diskette, into punched cards, or directly into the system.

Source Programs to Object Programs

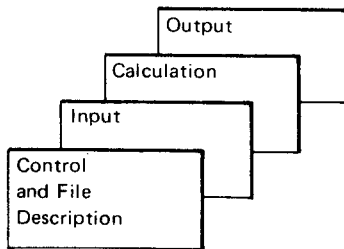
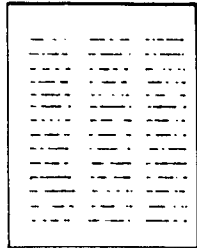
As we said earlier, the system understands only machine language. It cannot use a coded program (written in a programming language like RPG II) directly. Any program you write in RPG II must be translated into machine language. The translator is a program called a *compiler*. The RPG II Compiler program is available from IBM.

The compiler translates your RPG II specifications (source program) into machine language (object program). The translating it does is called *compilation*. Essentially, the compiler performs three functions during compilation:

- It determines what machine instructions are necessary for the system to perform the job described by your RPG II specifications.
- It translates your RPG II specifications into a machine language program.
- It assigns storage locations to program instructions and data.

Summary

Figure 2 illustrates the RPG II data processing concepts.



① Determine the requirements of your job. Define the input and printed output. Also decide what processing must be done in order to get the proper results.

② Write the source program by describing your job on the RPG II specification forms.

Figure 2 (Part 1 of 2). Summary of RPG II Data Processing Concepts

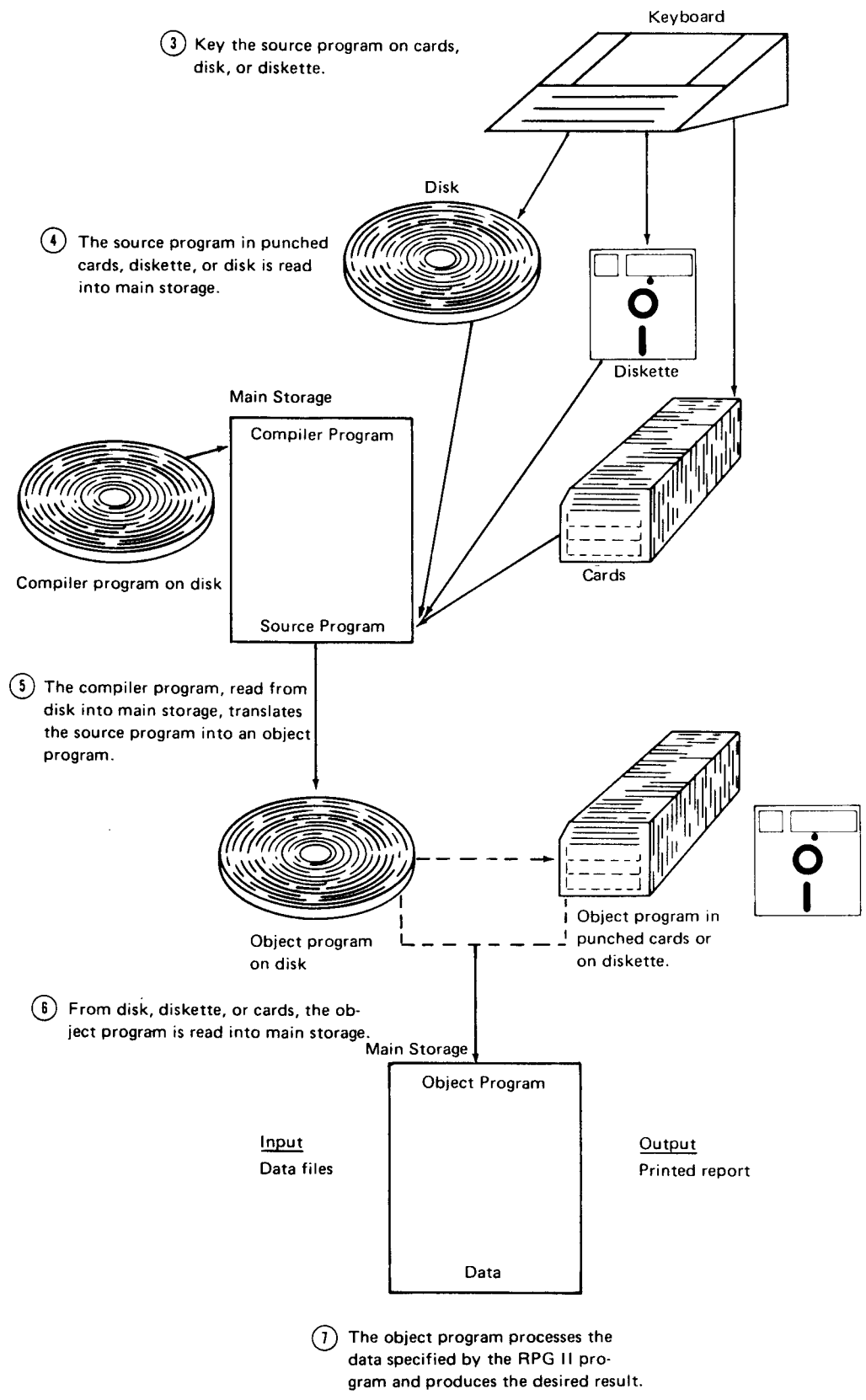


Figure 2 (Part 2 of 2). Summary of RPG II Data Processing Concepts

DATA PROCESSING TERMS AND PROGRAMMING AIDS

In the discussion of RPG II, you will find reference to data processing terms and programming aids. Figure 3 describes some basic programming terms. Figure 4 and Figure 5 are examples of some programming aids.

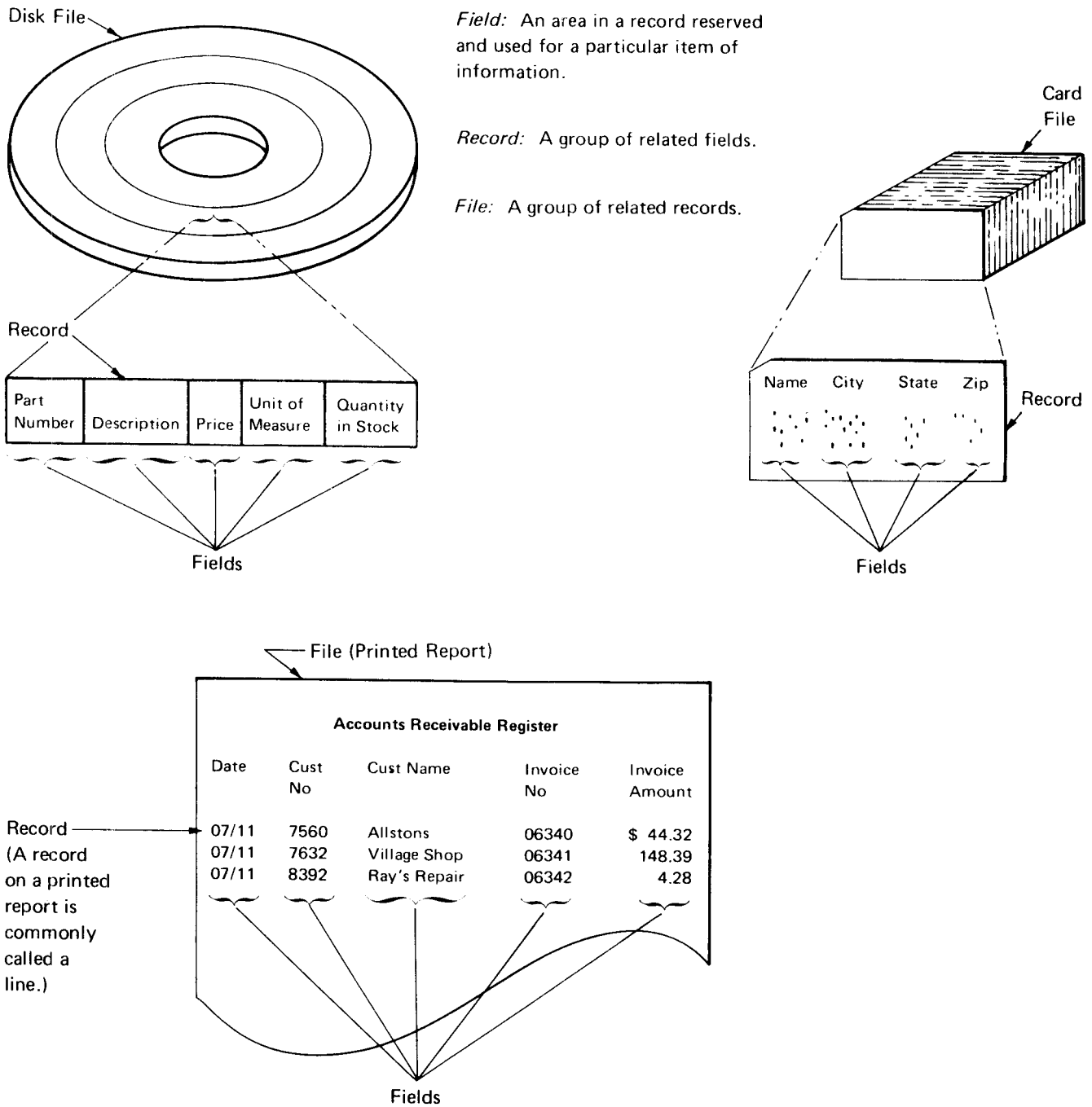
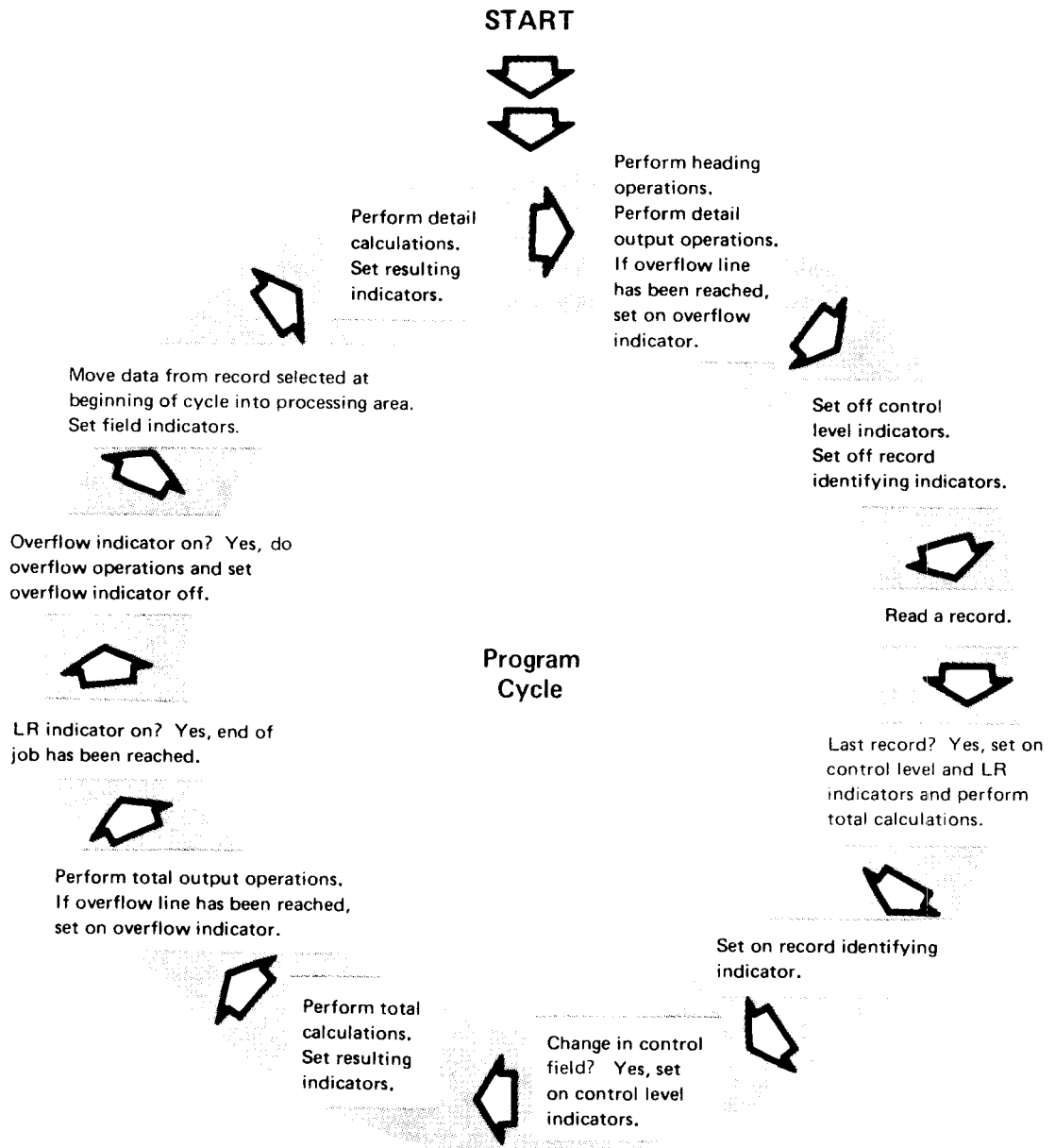


Figure 3. Basic Programming Terms

RPG II Program Cycle

When you do any job, you must do it in a particular order. The system must also do its job in a particular order. This logical order for the job is supplied by the RPG II compiler and/or your coding.

The logic the compiler supplies is called a *program cycle* (see Figure 6). The object program goes through this cycle of operations every time a record is processed. Depending on your specifications, the object program may or may not use a particular operation in the cycle. However, the program still goes through the complete program cycle every time. Because one program cycle is needed for each record read, many program cycles are required for every job.



Notes:

1. The program cycle shown gives the general order of the operations. There may be minor variations between this cycle and the detailed cycle discussed in the reference manual applicable to your system.
2. You do not need to memorize the program cycle. The cycle is only shown at this time to give you an idea of the cycle of the operations. The operations will be discussed in greater detail later.

Figure 6. Program Cycle

It is important that you know the order of the operations in the RPG II program cycle. This enables you to write specifications that will make correct use of the cycle. By knowing the order in which the operations in the cycle are performed, you can organize your program in an efficient manner and save yourself unnecessary coding.

In this chapter, the operations in the RPG II program cycle are explained a few at a time. You will learn:

- Which operations are used for a particular function.
- Which RPG II specifications you must write to use the function.

Writing Specifications for Input and Output Operations

One of the simplest jobs you can do on a system is read information from an input record, such as a card, then put that same information out, such as in the form of a printed report. No calculations are done.

Program Cycle Operations

To do this simple job, the system uses only the three most basic operations in the RPG II program cycle. Figure 7 shows these operations.

Notice that two operations are concerned with the basic requirements of a job: input (read a record) and output (detail output). The third operation is the movement of data inside the system.

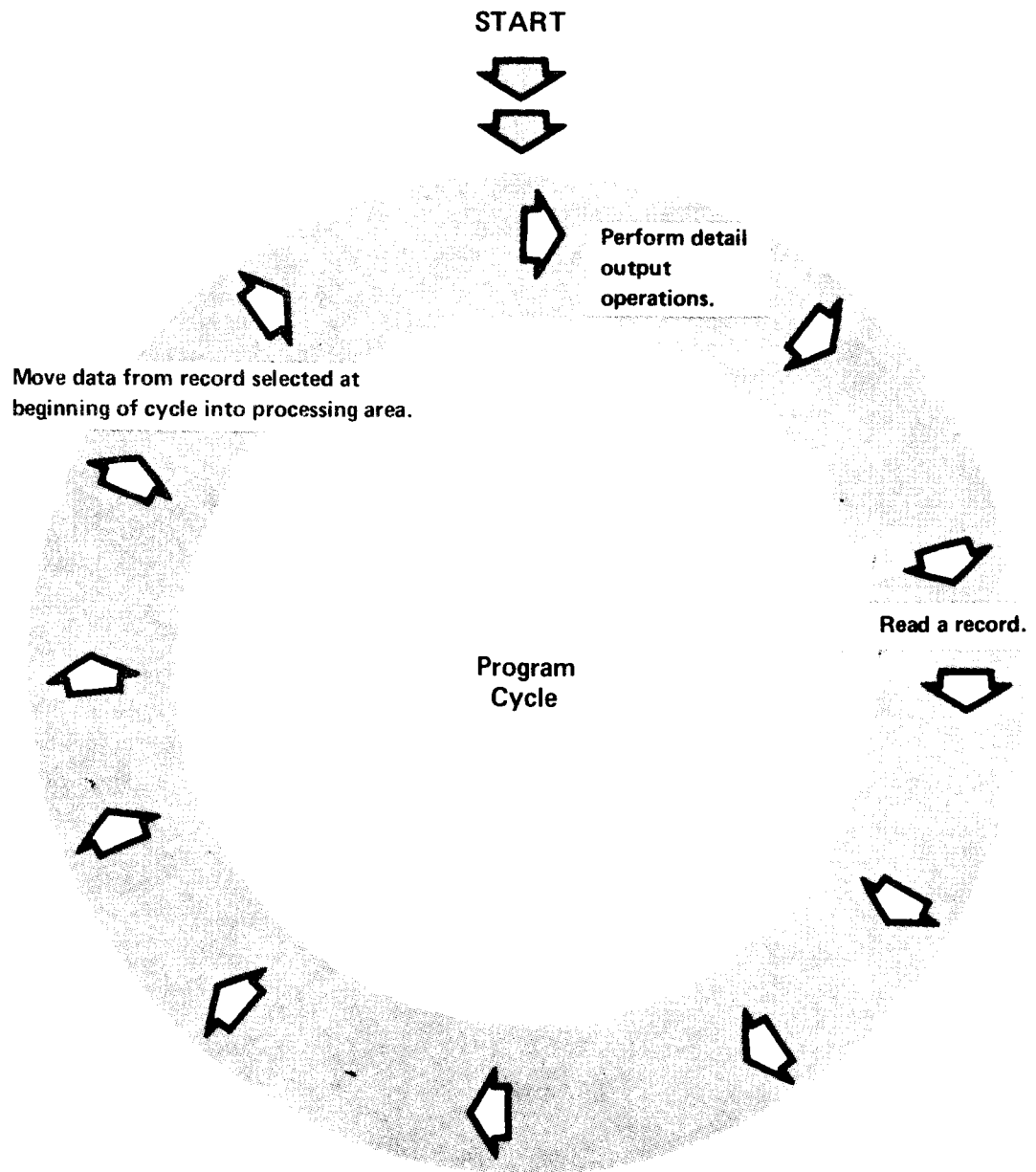


Figure 7. Three Basic Operations in the RPG II Program Cycle

Data read by an input device must be transferred to the system's processing unit before it can be used. Moving data is a mandatory operation done for every job. Because this operation is mandatory and is done exactly the same for every job, the compiler can supply instructions to do it.

When the program is executed, the program cycle is repeated over and over. All three operations are used for every record in the input file. The term *detail output*, in the cycle operation, means that the specified output operations are performed for every input record.

It may seem strange that detail output comes before a record is read. This occurs, however, so that headings can be printed on a report. If a job (such as this one) does not print headings, no information is printed during the first cycle.

To make proper use of these cycle operations, your specifications must describe the records in the input file and specify how the output records should be created. You must also indicate what devices are used in the job.

You must describe, on a separate line, every file used in your job. Many simple jobs require only one input and one output file. In the first jobs we discuss, therefore, we will use only one input and one output file.

File Names

Every file used in a job must be named. The name provides you and the compiler a means of identifying the file. During compilation, the compiler associates the file name with other characteristics of the file. Thus, you can refer to that file by name throughout your program and the compiler knows exactly which file you are referring to.

The compiler, however, recognizes file names only if they conform to these rules:

- A file name must be 1-8 characters long.
- The first character of a file name must be alphabetic. (The letters A-Z and the @, \$, and # signs are considered alphabetic characters.) The remaining characters in the name can be either alphabetic or numeric.
- Blanks must not appear between characters in the file name.
- No two files used in the same program can have the same name. (Because some RPG II compilers use only the first seven letters of an 8-letter file name, be certain, when using these systems, to make the first seven letters unique; for example, use TRANSACT and TRANFILE, not TRANFILA and TRANFILB.)
- The file name must begin in position 7 on the specifications form.

Line	Filename	Sequence
0 1	I TIMECRD	← Valid filename.
0 2	I LIST	← Invalid filename. Name must start in position 7.
0 3	I IN FILE	← Invalid filename. A blank must not be used between letters.
0 4	I PAYMASTR	← Valid filename.
0 5	I A	← Valid filename.
0 6	I IOUT	← Invalid filename. The first character of the name must be an alphabetic character.

It is a good practice to assign meaningful file names. Meaningful names indicate something about the file, such as the type of records in the file or the use of the file. Because file names can be no longer than eight characters, abbreviations may be necessary. But these too can be meaningful. For example, the abbreviation CUSTCHG might be assigned to an input file consisting of records for all customers having charge accounts.

Device Designation

You must also specify which devices your job will use for input and output. The ones you use, of course, depend upon the system you have, the devices you have, and your job.

To indicate the device used for the file you named, enter the RPG II code name for that device in positions 40-60. The name must begin in position 40.

Note: In the examples in this manual, you will see shading in the device name positions rather than actual RPG II code names for devices. Code names differ for each system. The RPG II reference manual for your system will tell you the code names that are applicable.

During compilation, the compiler associates the file name with the device name. When you use the same file name in the rest of your program, the system will know which device to use.

File Use

You must also describe how each file and its associated device is used in a program. Files can be used as either input or output. (Files with other uses will not be discussed in this manual.) If records are read from a file, the file is an input file. If a new file is created during the job, the new file is an output file. Printed reports are the only output files discussed in this manual.

You specify file use by placing either an I (input) or O (output) in position 15:

F															File Ty.	
Filename															File	
Line															IOUICD	File
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	2														I	
0	3														O	

← Input File

← Output File

File Designation

Position 16 is used to explain more about the use of input files. In this manual we are discussing the use of only one type of input file, the primary file. For this, you place a P in position 16. The designation of a file as a primary file in an RPG II program indicates that the RPG II program cycle will supply some of the program logic. For more information on primary files, and other types of input files, see the appropriate RPG II reference manual.

File Names

To tell the compiler which records you are describing, enter the name of the file containing them in positions 7-14. The name must be the same (and spelled exactly the same) as the one you assigned to the input file on the File Description Specifications form:

RPG INPUT SPECIFICATIONS

IBM International Business Machines Corporation

Program: *Stock Transaction Register* Keying Instruction: Card Electric Number: Page: 1 2

Programmer: *John Doe* Date: *1/17--* Key: Key:

I		F		File Type		Mode of Processing		Device	
Line	Filename	Line	Filename	File Designation	End of File	Record Address Type	Length of Key Field or of Record Address Field	Type of File	Organization or Additional Area
0.1	<i>TRANS</i>	0.1							
0.2		0.2							
0.3		0.3	<i>TRANS</i>	<i>IP</i>			<i>96</i>		

Note: A box labeled 'TRANS' is drawn around the filename in line 0.1 and line 0.3 of the 'F' section.

Field Names

To identify individual fields in the record, you must give each field a unique name. From information you placed on the File Description Specifications form the compiler determines the size of the storage area for each input record. The field names you supply on the Input Specifications form tell the compiler to divide this storage area into smaller sections so each can be addressed separately.

The rules for forming field names are as follows:

- The field name must be from 1-6 characters long.
- The first character must be alphabetic. Remaining characters can be either alphabetic or numeric.
- Blanks must not be placed between characters in a field name.
- The field name must begin in position 53 on the Input Specifications form.

Decimal Positions	Field Name	Control Loop (L1-L9)
53 54 55 56 57 58	NAME	
53 54 55 56 57 58	ORD NO	
53 54 55 56 57 58	ACCOUNT	
53 54 55 56 57 58	SHIP2	
53 54 55 56 57 58	%INCRS	

← Valid name.

← Invalid name. A blank cannot be used between letters in the name.

← Invalid name. A field name can be no longer than six characters.

← Valid name.

← Invalid name. The name must start with an alphabetic character (A-Z, #, \$, @).

Field Location

After you assign a field name, you must tell where the field is located in the record. This enables the compiler to associate the field name with the right information. To describe the field location, you specify the position in the record at which the field begins and ends. Starting position is specified in positions 44-47 (From); ending position is specified in positions 48-51 (To). When a field is only one character long, starting and ending position entries are the same. Field location entries can be easily determined from the record layout form:

		Record Identification Codes																				Field Location		Field Name	Control Level (1-10)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators		
Line	Filename	Sequence		Option (O, U, S)		Position		Position		Position		From	To	Plus	Minus	Zero or Blank														
1-4	5-13	O	S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
0.1	TRANS																													
0.2													3	8																
0.3													9	14																
0.4													15	17																
0.5													18	25																

The compiler also determines field length from the To and From entries. The compiler needs field length to determine how many storage positions to allow for each field. If the field locations you specify indicate a field length of 6, the compiler allows six positions in storage for the field.

Type of Data

To complete your description of the input fields, the compiler checks position 52. This position indicates whether data in each field is alphameric or numeric. A numeric field can contain only numbers; an alphameric field can contain numbers, letters, and special characters.

If position 52 is blank, the compiler assumes the field is alphameric. For numeric fields, position 52 must contain an entry. This entry indicates the number of decimal positions (digits to the right of the decimal point) in the field.

For a printed report, you make additional entries describing the format of the report; that is, make entries indicating the spacing and punctuation you want. These entries are discussed later under *Printed Reports*.

The printer spacing chart and record layout forms are useful when you are writing output specifications. The record layout form shows the organization of fields on a card or disk record; the printer spacing chart shows the format of printed records.

File Names

You indicate the output record you want created and the device you want to create the record by entering an output file name in positions 7-14. Make sure the name you enter is the same (and spelled exactly the same) as the name you entered on the File Description Specifications form for the output file:

RPG OUTPUT SPECIFICATIONS

IBM International Business Machines Corporation

Program Stock Transaction Register Keying Instruction Graphic Card Electro Number

Programmer John Doe Date 1/17/-- Key

Page 1 2

Space Skip Output Indicators Commas Zero Balances No Sign CR

File Description Specifications

Line	Form Type	Filename	File Type	Mode of Processing	Device
0 1	O	PRINT			
0 2	F	TRANS	IP		
0 3	F	PRINT			
0 4	F				

Note: A callout box highlights the 'PRINT' filename in line 0 1 and points to the 'PRINT' filename in line 0 3.

Record Type

Three different types of records which can be specified on the Output Specifications form are heading, detail, and total. You usually find all three types in a printed report.

Heading Records		ACCOUNTS RECEIVABLE TRANSACTION REGISTER							
		07/11/--					PAGE 01		
		DATE	CUST NO	CUSTOMER NAME	JOURNAL NO	INVOICE NO	CASH AMOUNT	INVOICE AMOUNT	JOURNAL AMOUNT
		07/11/--	759820	SOUND OF THE SEVENTIE		063420		\$ 46.23	
		07/11/--	633870	OLDE VILLAGE SHOPPE		063421		89.70	
		07/11/--	642990	PARAGON TV SALES		063422		20.30	
		07/11/--	122620	CANNIZONI STUDIOS		063422		129.76	
Detail Records		07/11/--	682030	RAYMONDS RAPID REPAIR			\$ 63.80		
		07/11/--	742950	SARATOGA VARIETY			29.72		
		07/11/--	014280	BAKER BRADLEY & CO			43.50		
		07/11/--	872060	UNIVERSITY ELECTRIC			97.75		
		07/11/--	883290	VILLAGE MUSIC & TV	07-036				\$18.23CR
		07/11/--	006280	ALLSTONS	07-037				10.70CR
Total Record		TOTALS					\$234.77*	\$285.99*	\$28.93CR*

Heading records are printed at the top of a page. They include report titles, column headings, or any other information needed to identify the kinds of information found in the report.

Detail records contain information about an individual item. Information in a detail record is often taken directly from an input record.

Total records are written after a group of detail records. They usually contain data that is the result of calculations on information in a group of detail records.

IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

Page 1 of 2

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Output Indicators	Field Name	End Position in Output Record
01	O	PRINT	D				*AUTO	39
02	O						CUST NO	53
03	O						ITEM	67
04	O						PRICE	77
05	O						QTY	

PRINTED REPORTS

When your output is a printed report, additional entries are needed on the Output Specifications form to make the report easy to read. Information must be neatly arranged in rows and columns with adequate space between items in a line and between lines.

Spacing

Your field location entries (positions 40-43) control space between fields, but to control spaces between lines you code positions 17-18 (Space).

You can have the printer single, double, or triple space between lines by entering the number 1, 2, or 3 in the appropriate positions. If you enter the number in Space Before (position 17), the printer spaces before printing the line; if you enter the number in Space After (position 18), the printer spaces after printing the line. You may enter numbers in both positions 17 and 18. By specifying three spaces before printing and three spaces after, you can have five blank lines between printed lines. The printer spaces three lines after printing a line, then spaces three lines before printing the next line. This next line is then printed six lines beyond the last line printed.

Editing

Editing is a means of punctuating numeric fields by adding decimal points, commas, and negative signs. Editing can also suppress the printing of leading zeros (in the number 00149, 00 are called leading zeros).

When a numeric field is read into storage, it contains no decimal point or commas; when an unedited numeric field is printed, it appears exactly as it is in storage. A large number printed without commas or decimals is hard to read. Furthermore, an unedited field may not be meaningful when printed because of the way the system stores negative numbers.

The system uses the last digit in a numeric field to indicate sign (plus or minus). If the field is minus, the system combines a minus sign with the last digit. When a negative number is printed unedited, the combination of digit and sign appears as a letter. For example, minus 6439 prints as 643R. On the other hand, a positive field has no sign (a numeric field that does not have a negative sign is assumed to be positive). A positive field, therefore, prints normally. Positive 6439 prints as 6439.

The compiler can provide instructions to edit in a number of ways. All you have to do is enter an edit code in position 38 of the Output Specifications form. Many codes are available, each indicating a different type of editing. Figure 8 shows the codes and the editing done for each. Figure 9 shows some examples of editing.

Note: When you edit a field, you often add characters to it. When printed, the edited fields require more space than they did on input records or in storage. When specifying end position for an edited field, always take into account the spaces needed for the punctuation that will be added. The printer spacing chart shows the amount of space needed for the edited field.

Edit Code	Commas	Decimal Point	Sign For Negative Balance			Zero Suppress	Print Out On Zero Balance
			No Sign	CR	- (Minus)		
1	Yes	Yes	No Sign			Yes	.00 or 0
2	Yes	Yes	No Sign			Yes	Blanks
3		Yes	No Sign			Yes	.00 or 0
4		Yes	No Sign			Yes	Blanks
A	Yes	Yes		CR		Yes	.00 or 0
B	Yes	Yes		CR		Yes	Blanks
C		Yes		CR		Yes	.00 or 0
D		Yes		CR		Yes	Blanks
J	Yes	Yes			-	Yes	.00 or 0
K	Yes	Yes			-	Yes	Blanks
L		Yes			-	Yes	.00 or 0
M		Yes			-	Yes	Blanks
X ¹							
Y ²						Yes	
Z ³						Yes	

¹The X code removes the plus sign of the field.

²The Y code is used for date fields. It suppresses only the leftmost zero and puts slashes in a three to six digit field according to the following pattern:
nn/n
nn/nn
nn/nn/n
nn/nn/nn

³The Z code removes signs and suppresses zeros.

Note: The edit codes shown in the first column are used in position 38 of the Output Specifications form to punctuate the field named on the same line. Only numeric fields can be edited. The decimal point is automatically inserted in the correct position.

Figure 8. Edit Codes

Field Length and Digits	1769532	02	00	000	041345	
	Positive Number—Two Decimal Positions	Negative Number—Two Decimal Positions	Zero—Two Decimal Positions	Zero—No Decimal Positions	Positive Number—Three Decimal Positions	
Edit Codes	1	17,695.32	.02	.00	0	41.345
	2	17,695.32	.02			41.345
	3	17695.32	.02	.00	0	41.345
	4	17695.32	.02			41.345
	A	17,695.32	.02CR	.00	0	41.345
	B	17,695.32	.02CR			41.345
	C	17695.32	.02CR	.00	0	41.345
	D	17695.32	.02CR			41.345
	J	17,695.32	.02-	.00	0	41.345
	K	17,695.32	.02-			41.345
	L	17695.32	.02-	.00	0	41.345
	M	17695.32	.02-			41.345
	X	1769532	OK	00	000	041345
	Y	Must be used with a 3 to 6-digit field.			0/0	4/13/45
	Z	1769532	2			41345

Note: This table shows the effect of editing on five different fields. It illustrates what will be printed by using each edit code on the fields.

Figure 9. Examples of Editing

JOB 1 (TRNREG): PRINTING A SIMPLE REPORT USING THE THREE BASIC CYCLE OPERATIONS

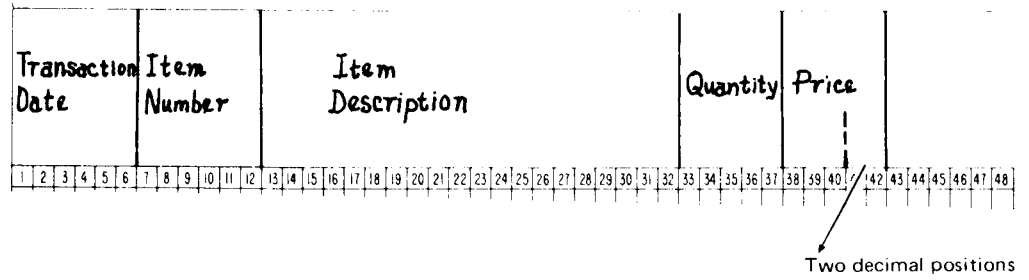
Job Definition

Print a report listing all items sold during a week. The selling of an item is known as a transaction, so the report is titled *Transaction Register*.

During the week, a transaction file is created. At the end of each day, transaction records are created from information obtained from order forms received during the day. To get the printed transaction report, you list the information from all input records on the printed report.

Job Requirements

Input: Sales transaction file consisting of 96-character records. The format of the input records is shown on this record layout form:



Output: A printed transaction register:

7/23/--	413010	CHOJ1 BOX 100A FLUSH	10	4.90
7/23/--	412146	CH148 BREAKER 15A	100	.89
7/23/--	411116	1500 TWIN SOCKET B	500	1.12
7/24/--	503029	MOTOR 1/2 HP 60 CYC	2	146.78
7/24/--	317802	TERMINAL CLIP	100	5.12
7/24/--	326917	TERMINAL BAR	100	4.12
7/24/--	411121	1506 SOCKT ADAPT BRN	400	.19
7/24/--	412997	CH173 BREAKER 3JA	60	1.15
7/24/--	413088	CH176 BREAKER 60A	40	1.15
7/24/--	411174	C151 SIL SWITCH BRN	200	1.16
7/24/--	413090	CHOJ5 BR BOX 150A	10	4.98
7/24/--	718326	FC903 FUSE 15A	200	.32

This printer spacing chart shows how the report is formatted:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	
D	XXXXXX	XXXXX	XXXXXXXXXXXXXXXXXXXX	XXXX	XXX.XX																																											
	(DATE)	(ITEM NUMBER)	(ITEM DESCRIPTION)	(QUANTITY)	(PRICE)																																											
	Single space all lines																																															

Program *Transaction Register* Keying Instruction Graphic Card Electro Number
 Programmer *John Doe* Date *1/10/--* Key

Page *03* of *12* Program Identification *TRMREG*

Line	Form Type	Filename	Type (HIDE)			Space			Skip			Output Indicators			Field Name	Edit Codes	End Position on Output Record	Commas	Zero Balances to Print	No Sign	CR	-	X	Remove Plus Sign	Date	Field Edit	Zero Suppress	
			A	D	N	Before	After	Next	And	And	Next	And	And	Next														
01	O	REPORT	D			1																						
02	O													DATE	Y	18												
03	O													ITEMNO		31												
04	O													DESC		60												
05	O													QTY	Z	70												
06	O													PRICE	3	79												

Output specifications describe how an output record will look. The output file is named first. D is entered in position 15 to indicate a detail line (one printed for every record read). A 1 in position 18 specifies single spacing. All fields to be printed are now listed, one per line, starting one line below entries describing the entire line. End Position, taken from the printer spacing chart, is given for each field. Again, according to the printer spacing chart, three fields are to be edited. DATE needs slashes (edit code Y), QTY must be zero suppressed (edit code Z), and PRICE must be zero suppressed and punctuated with decimals (edit code 3). Edit code 3 was chosen for the PRICE field instead of edit code 2 because PRICE, being a 5-position field with two decimals (xxx.xx), needs no commas.

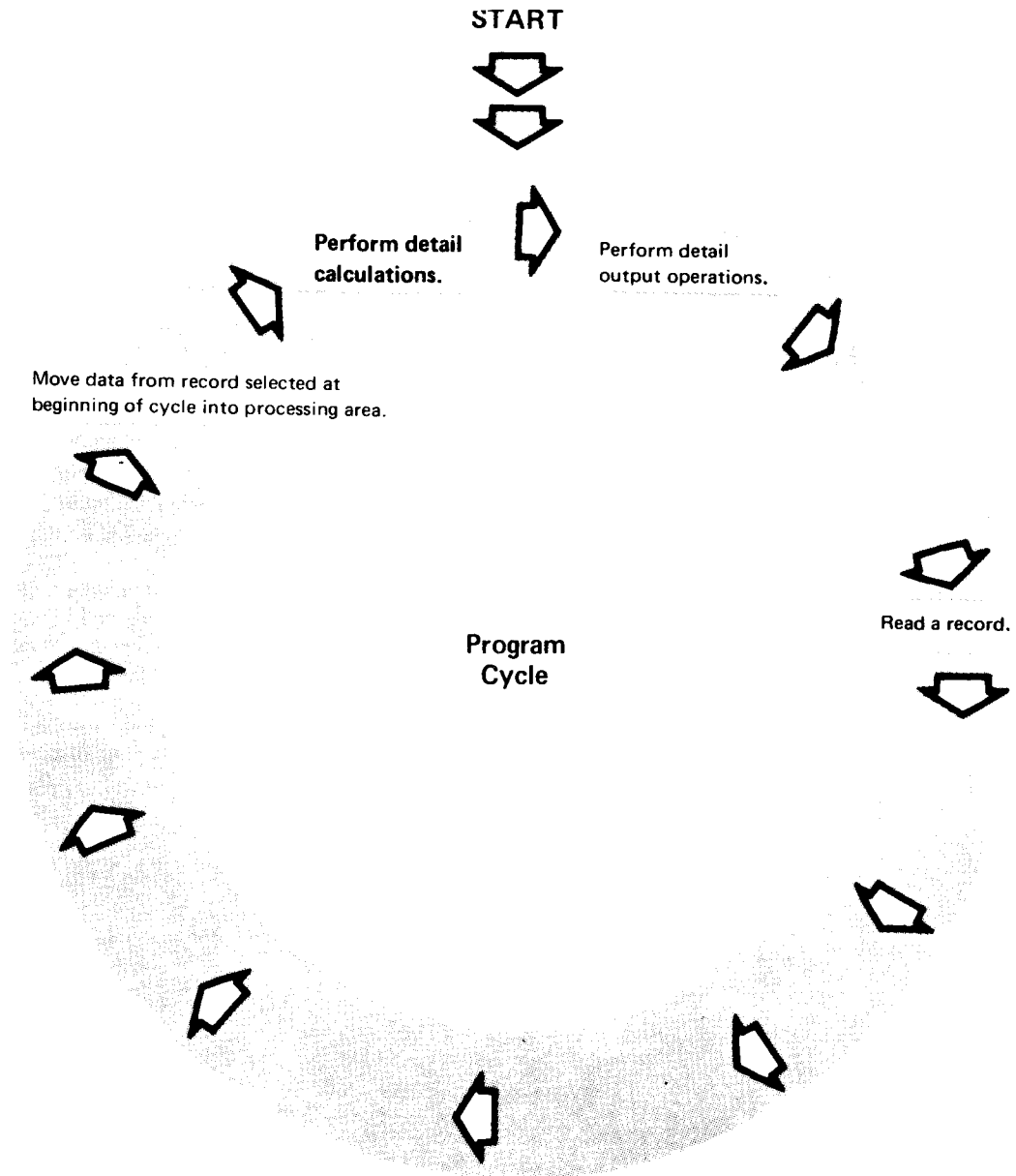
¹ The Record Identifying Indicator (entry 01 in positions 24 and 25) is needed to get correct output. It will be explained later under Job 5 (STOKST): Using Record Identifying Indicators to Process Different Record Types.

Writing Specifications for Calculation Operations

Most jobs require some processing. In RPG II, processing can include calculating, comparing, moving, or changing data. In this discussion we will consider only calculating (adding, subtracting, multiplying, and dividing).

Program Cycle Operations

When you specify a calculation operation related to each input record processed, you are adding one more operation to the basic program cycle: the detail calculation operation (see Figure 10).



Note: This is a basic program cycle showing the addition of detail operations. Because this is a detail operation, it is performed during every cycle for every record read.

Figure 10. Program Cycle for Detail Calculations

DESCRIBING TYPE OF OPERATION

To indicate the type of calculation operation, you enter one of the following operation codes in positions 28–32 on the Calculation Specifications form:

ADD (add)

SUB (subtract)

MULT (multiply)

DIV (divide)

DESCRIBING DATA TO BE USED

After you have specified the type of operation, you must identify the data to be used. If you specified ADD, for example, you must tell the system what to add. You do this by naming the fields to be used in positions 18–27 (Factor 1) and 33–42 (Factor 2).

Instead of naming a field in Factor 1 or Factor 2, you can enter a *constant*; that is, you can enter actual data instead of the name of a field containing the data. For example, you can enter either of the following:

500 Constant (actual data)

AMOUNT Name of a field containing data

Constants can be either numeric or alphameric, but for now we will discuss only numeric constants. The rules for using numeric constants are as follows:

- Constants can be up to ten numeric digits (0–9).
- Constants can have a sign and decimal point. The sign, if used, must be the leftmost character. The decimal point, if used, must be shown as part of the constant (4.12).
- The first character of the constant must be placed in the leftmost position of the Factor field.
- Constants cannot contain blanks.

The contents of a field can change during execution of a program, but constants do not. If you want to add, multiply, subtract, or divide the same number during every program cycle, you can use a constant:

Factor 1	Operation	Factor 2	Result Field		Comments
			Name	Length	
1	ADD	COUNT	COUNT		Add a constant 1 to COUNT at detail calculation time, thus providing a count of the records processed.
DIAM	MULT	3.14159	CIRCUM		Calculate the circumference of a circle by multiplying diameter by the constant 3.14159.
QTY	DIV	12	DOZEN		Convert a units quantity to a dozens quantity by dividing QTY by constant 12.

To the compiler, a constant is like a field name. During compilation, the compiler checks Factor 1 and Factor 2 for constants. If there are any, the compiler assigns a storage location for the constant and gives instructions to the computer to put the appropriate constant in that location at the beginning of job execution.

When you enter the fields in Factor 1 and Factor 2, be sure to consider their order because the specified operation may have an affect on the result:

ADD

Factor 2 is added to Factor 1 and the sum placed in the Result Field.

Factor 1		Operation	Factor 2		Result Field																									
					Name																									
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
AMT1		ADD	AMT2		TOTAMT																									
AMT2		ADD	AMT1		TOTAMT																									

Either line adds the two amount fields. The order of the fields makes no difference in addition.

SUBTRACT

Factor 2 is subtracted from Factor 1 and the difference placed in the Result Field.

Factor 1		Operation	Factor 2		Result Field																									
					Name																									
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
TOTAL		SUB	DEDUCT		DIFF																									
DEDUCT		SUB	TOTAL		DIFF																									

The order of factor 1 and factor 2 in subtract operations is important. The bottom line would not produce the desired result.

MULTIPLY

Factor 1 is multiplied by Factor 2 and the product placed in the Result Field.

Factor 1		Operation	Factor 2		Result Field																									
					Name																									
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
HOURS		MULT	RATE		GRSPAY																									
RATE		MULT	HOURS		GRSPAY																									

Either line multiplies the hours and rate to obtain the gross pay. The order of the fields makes no difference in multiplication.

DIVIDE

Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero.

Factor 1		Operation	Factor 2		Result Field																									
					Name																									
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
QTY		DIV	12		DOZEN																									
12		DIV	QTY		DOZEN																									

The order of the fields in divide operations is important. The bottom line will not convert a units quantity to dozens.

DESCRIBING THE RESULT FIELD

You must specify where you want the result of a calculation stored by naming that field in positions 43-48 (Result Field). The name you enter in the Result Field can be the name of a field already defined on the input or calculation specifications forms or a new field.

You would not need to name a new result field in these two situations:

- The contents of an input field are no longer required in your program, and the field is the correct size.
- The contents of an input field or a result field already defined on the Calculation Specifications form is to be replaced with a new value.

Result Field Length

When you name a result field, make sure you specify one large enough to hold the results. Always consider the length of the fields involved in the operations. For example, if you are adding a two-position field to a three-position field, you must determine the largest result you could possibly have:

$$\begin{array}{r} 999 \\ + 99 \\ \hline 1098 \end{array}$$

Because there are four digits in this result, you would specify at least 4 as the result field length.

If this calculation would occur many times in your program, as in a running total, you would probably need a result field length larger than 4. It is up to you to determine the largest field length needed; failure to specify a large enough result field can mean a loss of data.

Decimal Positions

For a new result field, be certain to place an entry in position 52. If the new field is to be numeric but contains no digits to the right of the decimal point, enter a zero. Remember, this entry indicates type of field (numeric or alphameric) as well as decimal position. If the result field is not specified as numeric by an entry in position 52, the compiler will not provide instructions for the calculation operation.

Half-Adjusting Results (Rounding)

In RPG II, rounding results is called *half-adjusting*. When the digit to the right of the last digit you want to keep is greater than 4, 1 is added to the last digit. The number 3.14159 rounded to four decimal positions becomes 3.1416. The same number rounded to two decimal positions is 3.14.

JOB 2 (TRNREG): DOING SIMPLE CALCULATIONS

Job Definition

Print a report listing all sales transactions for a week. This report is similar to the report created in Job 1. The only difference is the addition of the last column on the report which is the sales amount per item. Sales amount (quantity sold times item price) is not found in the input record and must, therefore, be calculated.

Job Specifications

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21 0092 GM/050*
Printed in U.S.A.

IBM International Business Machines Corporation

Program	Transaction Register		Keying	Graphic	Card Electro Number
Programmer	John Doe		Instruction	Key	
Date	1/10/--				

Page 01 of 12
Program Identification TRWREG

Control Specifications

Line	Form Type	Size to Complete	Object Output Linking Options	Size to Execute	Debit	MFCM Shading Sequence	Reserved	Date Format	Date Exit	Invert Print	Number Of Print Positions	Alternate Coding Sequence	Reserved	Validity	Sign Handling	IP Form Position	COB Area	Dimension
01	H																	

The same files are used for Job 1. Therefore, the File Description entries are the same.

File Description Specifications

Line	Form Type	Filename	File Type	File Designation	End of File	Sequence	File Format	Block Length	Record Length	Mode of Processing	Length of Key Field or of Record Address Field	Record Address Type	Type of File Organization or Additional Area	Overflow Indicator	Key Field Starting Location	Device	Symbolic Device	Label S/N/E/M	Name of Label Exit	Extent Exit for DAM	Storage Index	File Abandon/Unordered	Number of Tracks for Cylinder Overflow	Number of Exits	Tape Rewind	File Continuation U/I
02	F	TRANS	IP						96							PRINTER										
03	F	REPORT	O						96																	
04	F																									

RPG INPUT SPECIFICATIONS

GX21 0094 GM/050*
Printed in U.S.A.

IBM International Business Machines Corporation

Program	Transaction Register		Keying	Graphic	Card Electro Number
Programmer	John Doe		Instruction	Key	
Date	1/10/--				

Page 02 of 12
Program Identification TRWREG

Line	Form Type	Filename	Sequence	Record Identification Codes	Field Location	Field Indicator
01	I	TRANS	01	1 DATE	60	
02	I			7 ITEMNO	12	
03	I			13 DESC	32	
04	I			33 QTY	37	
05	I			38 PRICE	42	
06	I					
07	I					
08	I					
09	I					
10	I					
11	I					
12	I					
13	I					
14	I					
15	I					
16	I					
17	I					
18	I					
19	I					
20	I					

Input specifications are the same as for Job 1 because the same input file is used.

RPG CALCULATION SPECIFICATIONS

Program Transaction Register Keying Instruction _____ Graphic _____ Card Electro Number _____
 Programmer John Doe Date 1/10/-- Key _____

Page 03 of 1 2 Program Identification TRMREG

Line	Form Type	Control Level (LOL, LR, SR, AN, OR)	Indicators			Factor 1	Operation	Factor 2	Result Field		Resulting Indicators	Comments
			And	And	And				Name	Length		
01	C					QTY	MULT	PRICE	EXTCST	72		
02	C											
03	C											
04	C											
05	C											
06	C											
07	C											
08	C											
09	C											
10	C											
11	C											
12	C											
13	C											
14	C											
15	C											
16	C											
17	C											

This specification tells the system what calculation to perform. QTY must be multiplied by PRICE. QTY and PRICE are fields from the input record and are described on the Input Specifications form. Notice, however, that a new field, EXTCST, is created to hold the result. We needed a new result field because we could not use any of the fields described on the Input Specifications form. If we had used one of them, we would lose information needed for printing the detail line.

Any new field must be defined by describing field name, field length, and decimal positions. We chose the name EXTCST. Any valid name could be used. According to the printer spacing chart, the EXTCST field is 7 positions long with two decimal positions. We, therefore, used these figures when defining field length and decimal positions.

RPG OUTPUT SPECIFICATIONS

Program Transaction Register Keying Instruction _____ Graphic _____ Card Electro Number _____
 Programmer John Doe Date 1/10/-- Key _____

Page 04 of 1 2 Program Identification TRMREG

Line	Form Type	Filename	Type Indicators			Space	Skip	Output Indicators			Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word	Commas	Zero Balances to Print	No Sign	CR	X - Remove Plus Sign	Y - Date	Z - Field Edit	- Zero Suppress	
			OR	AND	OR			And	And	And													
01	O	REPORT	D	I							*AUTO												
02	O										DATE	Y	18										
03	O										ITEMNO		31										
04	O										DESC		60										
05	O										QTY	Z	70										
06	O										PRICE		79										
07	O										EXTCST		90										
08	O																						
09	O																						
10	O																						
11	O																						
12	O																						
13	O																						
14	O																						
15	O																						
16	O																						
17	O																						
18	O																						
19	O																						
20	O																						

This Output Specifications form differs from the one in Job 1 by only one entry. The field EXTCST was added because it is to be included in the output line. EXTCST is to be edited with commas. Thus, we use edit code 1.

CONTROL LEVEL INDICATORS

Control level indicators are used when you want to calculate and print totals. Nine different indicators can be used (L1 through L9), allowing as many as nine different levels of totals in the same program. The control level indicators tell the program two things:

- When totals should be calculated.
- Which calculations and output operations are total operations.

A control level indicator in positions 59 and 60 next to an input field specified on the Input Specifications form determines when totals should be calculated and printed. This input field is called a *control field*. Whenever the contents of the control field changes, a *control break* occurs. A control break turns on the control level indicator assigned to the control field and all lower control level indicators. The system performs all calculations and output operations (total operations) that are conditioned by these control level indicators. If a control break causes L3 to be turned on, L1 and L2 are turned on. This allows control over several levels of totals and subtotals. Examples are daily, weekly, and monthly totals.

In Figure 11, the records are grouped by date and a total of the sales amount field should be printed when the contents of the input record date field changes.

Program Cycle Operations

Figure 12 shows the program cycle operations associated with control level indicators. The system can do calculations and output operations at two different times in one cycle: at detail time and at total time. Total operations associated with control level indicators are not done in every cycle; they are done during the cycle in which the control field changes.

After a record is read, the program determines whether the control field in the record just read is different than the control field in the previous record. If it is, a control break occurs and the control level indicator you specified is set on. When the indicator is on, it means that all records in the control group have been read and total operations can be performed. Control level indicators remain on through the detail calculations and output processing of the record that caused the control break. They are then set off before the next record is read.

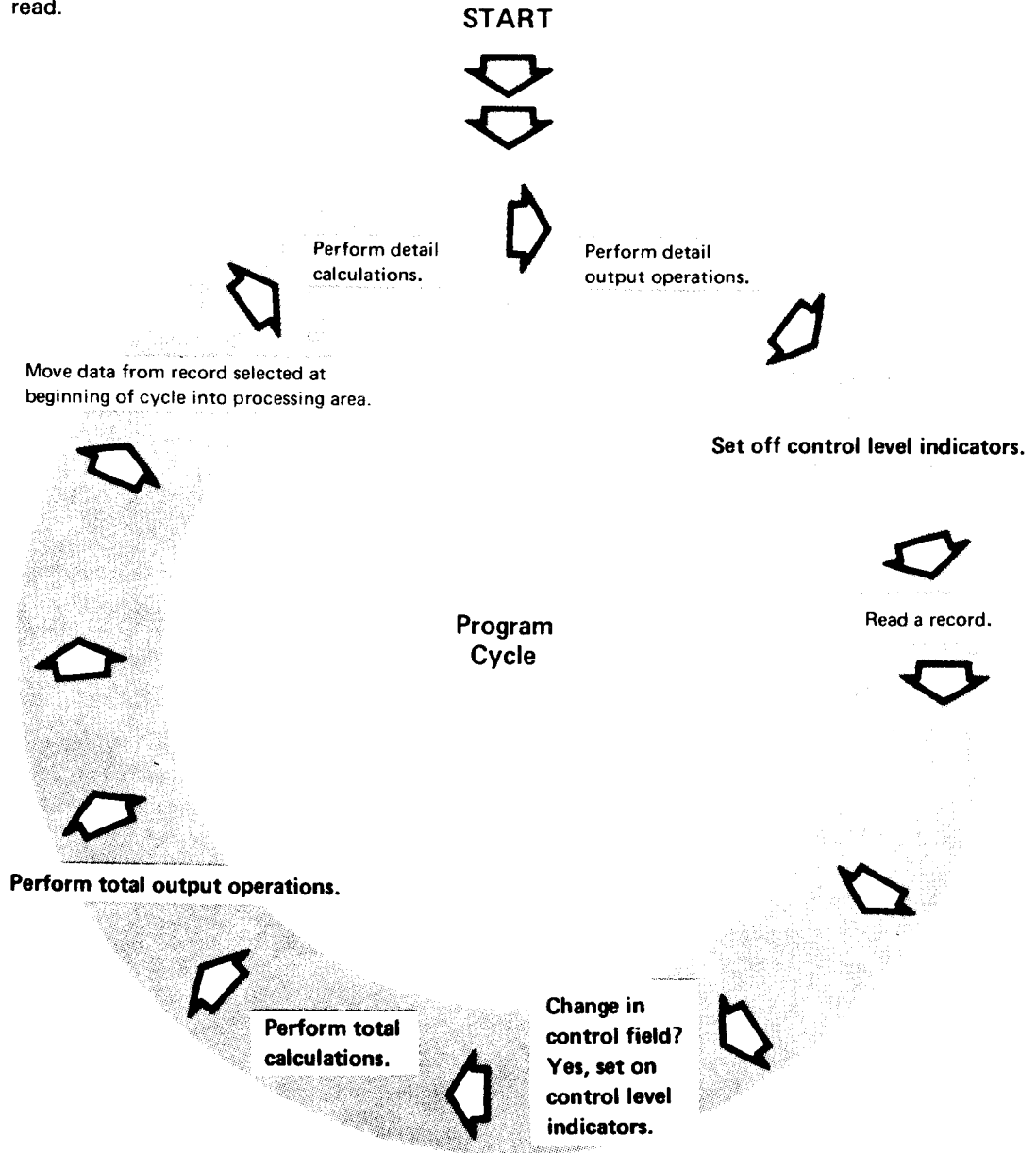


Figure 12. Program Cycle Operations for the Control Level Indicators

RPG INPUT SPECIFICATIONS

GX21 9094 UM/050*
Printed in U.S.A.

IBM International Business Machines Corporation

Program	Keying Instruction	Graphic	Card Electro Number
Programmer	Date	Key	

Page 1 of 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Record Identification Codes									Field Location		Field Name	Field Indicators				
			1	2	3	4	5	6	7	8	9	From	To		Plus	Minus	Zero or Blank		
01	I	INPUT																	
02	I																		
03	I																		
04	I																		
05	I																		

RPG CALCULATION SPECIFICATIONS

GX21 9093 UM/050*
Printed in U.S.A.

IBM International Business Machines Corporation

Program	Keying Instruction	Graphic	Card Electro Number
Programmer	Date	Key	

Page 1 of 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level (L0, L1, L2, L3, L4, L5, L6, L7, L8, L9)	Indicators			Factor 1	Operation	Factor 2	Result Field		Resulting Indicators	Comments
			And	And	And				Name	Length		
01	C					QTY	MULT	PRICE	EXTCST	72		
02	C					EXTCST	ADD	DAYTOT	DAYTOT	72		
03	C	L1				DAYTOT	ADD	FINCST	FINCST	102		
04	C											
05	C											
06	C											
07	C											
08	C											
09	C											
10	C											
11	C											
12	C											

Use a control level indicator in positions 7-8 to show that a total operation is to be done only when a control break occurs. The operations on line 01 and line 02 are done during the detail processing of each input record. The ADD operation in line 03 is a total operation that will be done when L1 is on; that is, when the DATE field changes.

RPG OUTPUT SPECIFICATIONS

GX21 9090 UM/050*
Printed in U.S.A.

IBM International Business Machines Corporation

Program	Keying Instruction	Graphic	Card Electro Number
Programmer	Date	Key	

Page 1 of 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Space	Skip	Output Indicators		Field Name	End Position in Output Record	P/B/L/R	Constant or Edit Word
					And	And				
01	O	OUTPUT					DATE	6		
02	O						QTY	25		
03	O						PRICE	35		
04	O						EXTCST	45		
05	O									
06	O				L1		DAYTOT	45		
07	O									
08	O									
09	O									
10	O									

The T in position 15 indicates which output records are total records. Every total record should also have a control level indicator specified to tell the system when to do the operation. The output operation described on lines 06 and 07 is done only when L1 is on.

You can specify up to three different indicators on a line on the Output Specifications form. If you are using only one indicator, you can enter it in any one of the three positions. The control level indicators specified on this form can be used to condition an entire output record or only certain fields in the output record.

IBM		RPG OUTPUT SPECIFICATIONS										GX21 9090 UM/050*																						
Programmer		Date		Keying Instruction		Graphic Key		Card Electro Number		Page 1 of 2		Program Identification 75 76 77 78 79 80																						
Line	Form Type	Filename	Type (H/DT/RE)	Space	Skip	Output Indicators			Field Name	End Position in Output Record	Field Length	Constant or Edit Word																						
				Before	After	Not	Not	Not				Commas	Zero Balances to Print	No Sign	CR	-	X	Remove Plus Sign	Date	Field Edit	Zero Suppress													
01	O	OUTPUT	D									Yes	Yes	1	A	J	Y																	
02	O								DATE	Y	6	Yes	No	2	B	K																		
03	O								QTY	Z	25	No	Yes	3	C	L																		
04	O								PRICE	3	35	No	No	4	D	M																		
05	O								EXTCST	1	45																							
06	O		T						DAYTOT	1	45																							

A control level indicator specified on the same line as the field name ① indicates that the field should be written only when the control level indicator is on. Indicator L1 will be on at detail output time only for the first record of each group. This record is the same record which caused the control break and allowed the total operations for the previous group to take place. DATE will print only on the first detail line of each group. However, a control level indicator specified on the same specification line as the line type entry in position 15 ② indicates that the entire line should be written when the control level indicator is on.

Using the Blank-After Specification

In RPG II, you can set fields in storage to blanks (in the case of alphameric fields) or zeros (in the case of numeric fields) after they have been written out. You do this by entering a B in position 39 of the Output Specifications form.

This is a particularly useful feature when you are doing total operations. It allows you to use the same field over and over for accumulating and printing totals. For example, you could use a numeric field to accumulate totals for a particular group of records. After the totals are accumulated and printed for that group, you can use the same numeric field to accumulate the totals for the next group of records. To do this, place a B in position 39 for the total field. If you do not place a B in position 39, the totals for the second group of records would be added to the totals for the first group of records.

JOB 3 (TRNREG): USING CONTROL LEVEL INDICATORS TO CALCULATE AND PRINT TOTALS

Job Definition

Print a weekly sales transaction report that lists all daily transactions and gives the total sales for each day. This report is similar to the reports produced in Jobs 1 and 2. All items sold each day are listed. Item number, item description, quantity sold, unit price, and sales amount (quantity times unit price) are included for each item. The date is printed only for the first transaction encountered that has a new date. The total sales amount for a day is printed after all transactions for that day have been recorded.

Job Requirements

Input: Sales transaction file consisting of 96-character records. The records are arranged in ascending order by date. The format of the input records is shown on the record layout form:

Transaction Date	Item Number	Item Description	Quantity	Unit Price
1 2 3 4 5 6	7 8 9 10 11	12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32	33 34 35 36 37	38 39 40 41 42 43 44 45 46 47 48

Two decimal positions

Processing:

- Multiply quantity times unit price to find sales amount.
- Find total of all item sales per day.

Output: A printed transaction register:

Group Indication	7/23/--	413010	CH001 BDX 100A FLUSH	10	4.90	49.00
		412146	CH148 BREAKER 15A	100	.89	89.00
		411116	1500 TWIN SOCKT B	500	1.12	560.00
						698.00
	7/24/--	503029	MOTDR 1/2 HP 60 CYC	2	146.78	293.56
		317802	TERMINAL CLIP	100	5.12	512.00
		326917	TERMINAL BAR	100	4.12	412.00
		411121	1506 SOCKT ADAPT BRN	400	.19	76.00
		412997	CH173 BREAKER 30A	60	1.15	69.00
		413088	CH176 BREAKER 60A	40	1.15	46.00
		411174	C151 STL SWITCH BRN	200	1.16	232.00
		413090	CH005 BR BDX 150A	10	4.98	49.80
		718326	FL803 FUSE 15A	200	.32	64.00
						1,754.36

This printer spacing chart shows how the report is formatted:

	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
1																																																																																																															
2																																																																																																															
3																																																																																																															
4																																																																																																															
5																																																																																																															
D 6																																																																																																															
7																																																																																																															
8																																																																																																															
9																																																																																																															
T 10																																																																																																															
11																																																																																																															
12																																																																																																															
13																																																																																																															
14																																																																																																															

Job Specifications

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092 UM/050*
Printed in U.S.A.

IBM International Business Machines Corporation

Program <i>Transaction Register</i>	Keying Instruction	Graphic	Card Electro Number
Programmer <i>John Doe</i>	Date <i>1/10/--</i>	Key	

Page 01 of 2 Program Identification TRWREG

Control Specifications

Line	From Type	Size to Compile	Object Output Listing Options	Size to Execute	Debug	MFCM Stacking Sequence	Reserved	Date Format	Date Edit	Insert Print	Reserved	Number of Print Positions	Altstreams Collating Sequence	Reserved	Inquiry	Reserved	Sign Handling	IP Forms Position	Indicator Setting	File Translation	Jump MFCU Zeros	Nonprint Characters	Reserved	Table Load Hgt	Shared I/O	Field Print	Formatted Dump	RPG to RPG II Conversion	Number of Formats	S 3 Conversion	Subprogram	CICS DLI	Refer to the specific System Reference Library manual for actual entries	
01	H																																	

No new entries are made on this form.

File Description Specifications

Line	From Type	Filename	File Type	File Designation	Sequence	File Format	Block Length	Record Length	Mode of Processing	Record Address Type	Device	Symbolic Device	Name of Label Exit	Storage Index	File Addition/Unordered
02	F	TRANS	OP					96	L/R		PRINTER				
03	F	REPORT	OP					96	L/R						
04	F														
05	F														

RPG INPUT SPECIFICATIONS

GX21-9094 UM/050*
Printed in U.S.A.

IBM International Business Machines Corporation

Program <i>Transaction Register</i>	Keying Instruction	Graphic	Card Electro Number
Programmer <i>John Doe</i>	Date <i>1/10/--</i>	Key	

Page 02 of 2 Program Identification TRWREG

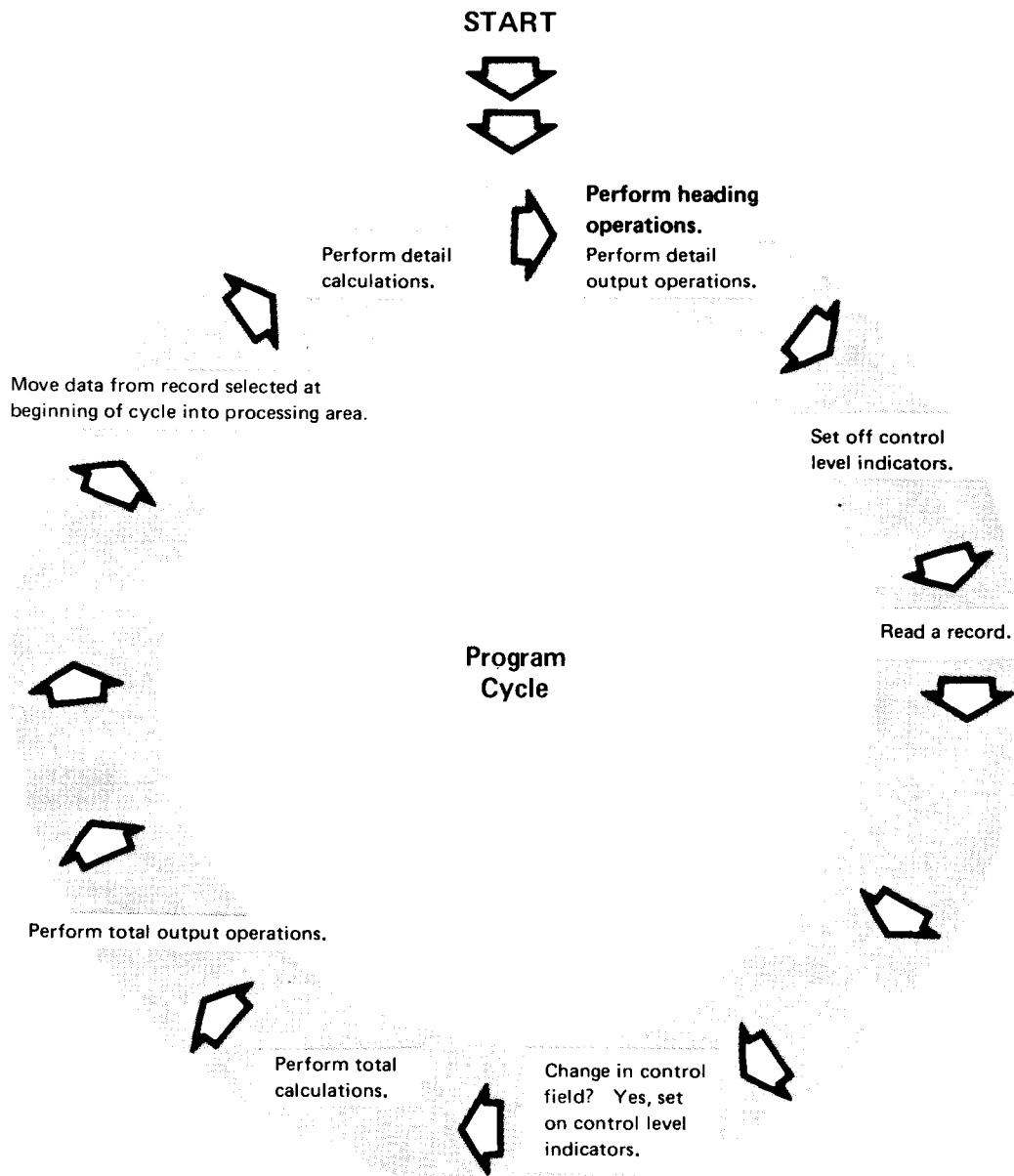
Line	From Type	Filename	Sequence	Record Identification Codes	Field Location	Field Name	Field Indicators
01	I	TRANS	01	1 DATE	60 DATE	L1	
02	I			2 ITEMNO	12 ITEMNO		
03	I			3 DESC	32 DESC		
04	I			4 QTY	37 QTY		
05	I			5 PRICE	42 PRICE		

Input fields are described as before. In this job, totals must be accumulated and printed. You know that totals must be printed whenever a record with a different date field is read. The date field determines when total operations should be done. The date field is the control field and must be specified as such. This is done by the L1 entry in positions 59-60.

Program Cycle Operations

One operation in the program cycle is concerned with the 1P indicator (see Figure 13). The 1P indicator is automatically set on at the beginning of every job, so the first operation taken by the system is to print any output record conditioned by 1P. After this is done, the first record is read and the program cycle operations are executed in order.

Headings conditioned by 1P are printed only once—at the beginning of the job on the first page of the report. Any heading records that are not conditioned by 1P are handled in the same way as detail records. This means that they will be printed along with detail records in every cycle.



Note: The first operation in the first program cycle concerns output operations conditioned by the first page (1P) indicator.

Figure 13. Program Cycle Operations for 1P Indicators

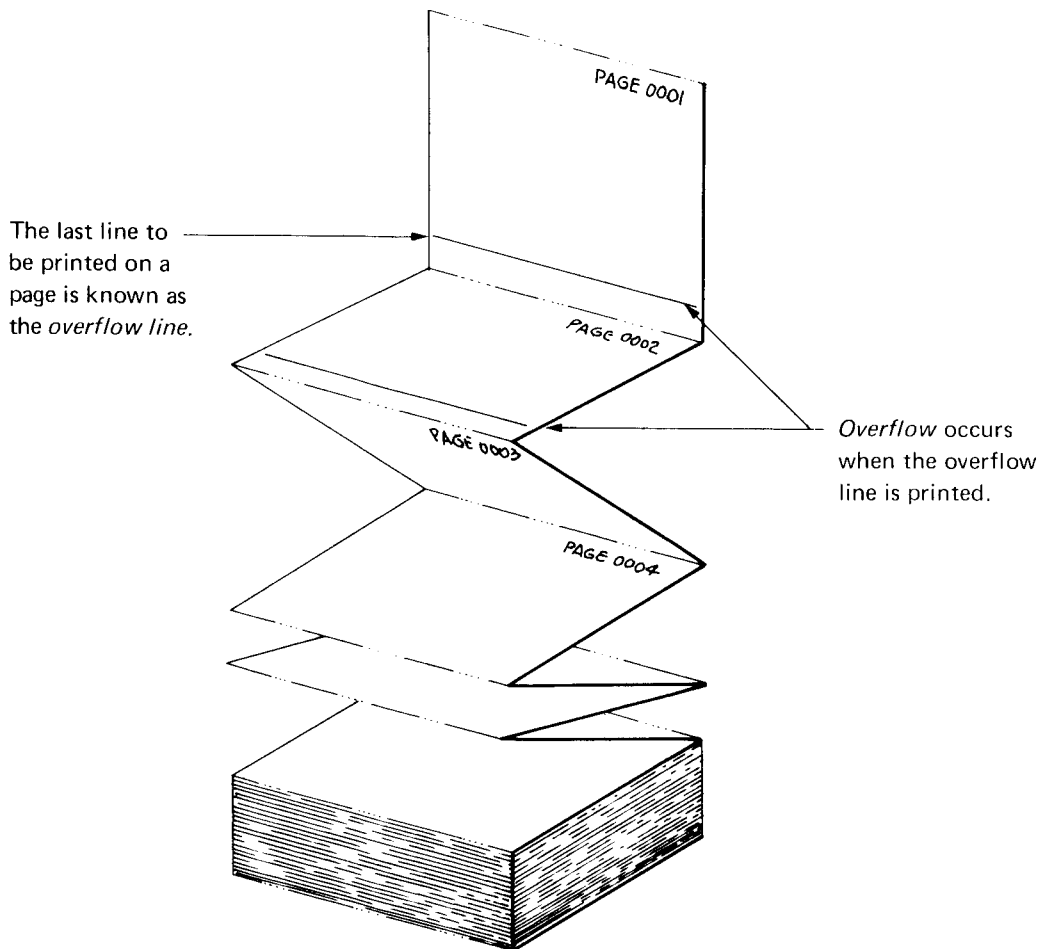
OVERFLOW INDICATORS

You use overflow indicators to:

- Print headings on every page but the first page of a report (the 1P indicator allows headings to be printed on the first page).
- Control where printing begins and ends on a page.
- Advance forms from one page to the next (provided a skip specification is also used).

To understand how overflow indicators work, you must know how the concept of overflow is defined in RPG II:

- Overflow—Lines that remain to be printed after a page is full.
- Overflow handling—Advancing forms to a new page after the last line has been printed on the current page.
- Overflow line—The last line to be printed on a page.
- Overflow page—The new page to be printed when overflow occurs.



Printers use continuous forms (a series of pages divided by perforations). Overflow handling refers to the means of advancing forms from one page to the next.

Overflow can be handled automatically by the system or through specifications you write.

Note: This discussion assumes that you are using standard 66-line forms in the printer.

Printing always begins on line 06 (assuming the operator positions the first page properly) and ends with line 60. Overflow occurs after line 60 is printed; that is, forms advance to line 06 of a new page.

When overflow is handled automatically, heading lines can be printed on the first page if 1P is used. No instructions are provided, however, to print headings on overflow pages. Heading lines are considered to be detail lines and will print on each detail cycle if the conditions placed on printing are met.

When you do not want overflow handled automatically, you can specify on the coding forms how you want it handled, using overflow indicators:

File Description Specifications

Line	Form Type	File Description Specifications																				
		File Type						Mode of Processing						Device	Symbolic Device	Name of Label Exit		Extent Exit for DAM		File Addition/Unordered		
		File Designation	End of File	Sequence	File Format	Block Length	Record Length	L/R	Length of Key Field or of Record Address Field	Record Address Type	Type of File Organization or Additional Area	Overflow Indicator	Key Field Starting Location			Extension Code E/L	Option	Entry	Storage Index	Continuation Lines	Number of Tracks for Cylinder Overflow	Number of Extents
0 2	F																					
0 3	F																					
0 4	F																					
0 5	F																					
0 6	F																					

RPG OUTPUT SPECIFICATIONS

IBM International Business Machines Corporation

GX21-9090: UM/060*
Printed in U.S.A.

Program	Keying Instruction	Graphic	Card Electro Number	Page 1 2 of	Program Identification
Programmer	Date	Key			75 76 77 78 79 80

Line	Form Type	Filename	Output Indicators										Field Name	End Position in Output Record	P/B/L/R	
			Space	Skip	And	And	And	And	And	And	And	And				
0 1	O															
0 2	O															
0 3	O															
0 4	O															
0 5	O															
0 6	O															
0 7	O															
0 8	O															
0 9	O															
1 0	O															
1 1	O															

Assign an overflow indicator to the printer at ①, then use it on the Output Specifications form at ② to show which operations must be done when overflow occurs.

Program Cycle Operations

Figure 14 shows operations in a program cycle in which overflow indicators are used.

The program sets on the overflow indicator you assigned whenever the overflow line is passed. By setting the overflow indicator on, the program remembers that overflow has occurred. As you can see in Figure 14, overflow indicators can be set on at one of two times: at detail time when a detail record prints on the overflow line or at total time when a total record prints on the overflow line. Notice that the only time a check is made to see if the overflow indicator is on is right after total output. If the overflow indicator is on, overflow operations are done in this order:

1. Print any total lines conditioned by the overflow indicator.
2. Skip to new page, provided a skip specification was made on a line conditioned by the overflow indicator.
3. Print all heading and detail lines conditioned by the overflow indicator.

If multiple detail or total lines are to be printed in a single cycle, printing may occur past the designated overflow line. This is because all detail and total printing for a single cycle is completed before overflow operations occur.

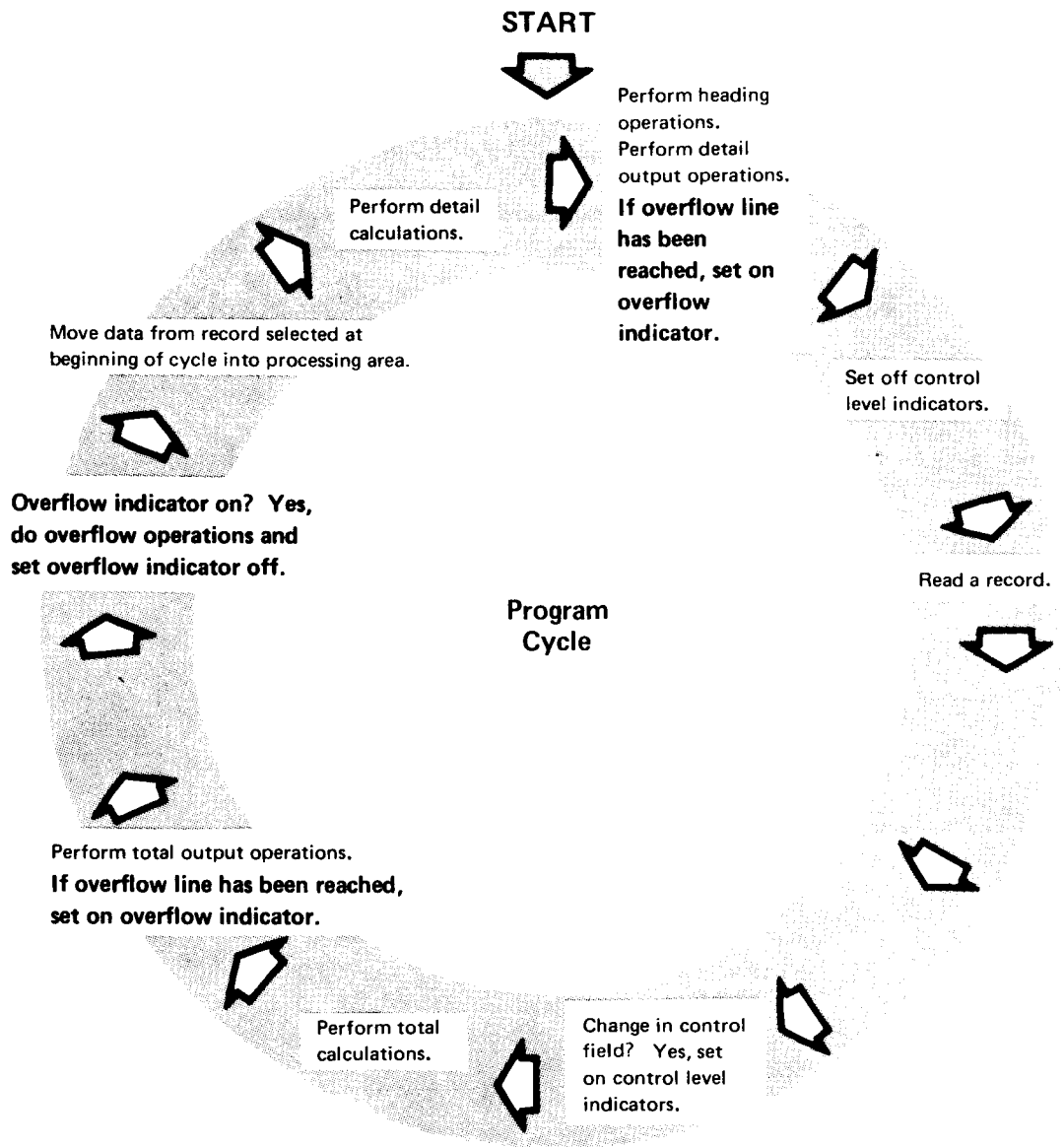


Figure 14. Program Cycle Operations for Overflow Indicators

Program Cycle Operations

Figure 16 shows the operations in the program cycle associated with the last record indicator. RPG II is set up so that it senses an end-of-file record containing some identifying information to indicate end of the data file. Each data file has an end-of-file record.

Whenever a record is read, the program checks to see if all records in a file have been processed. If all records in all of the files have been processed, the program sets on all control level indicators L1-L9. It also sets on the LR indicator to indicate that all records have been processed. All total operations (those conditioned by LR and L1-L9) are performed. After total operations have been done, the program checks to see if LR is on. If it is, processing stops.

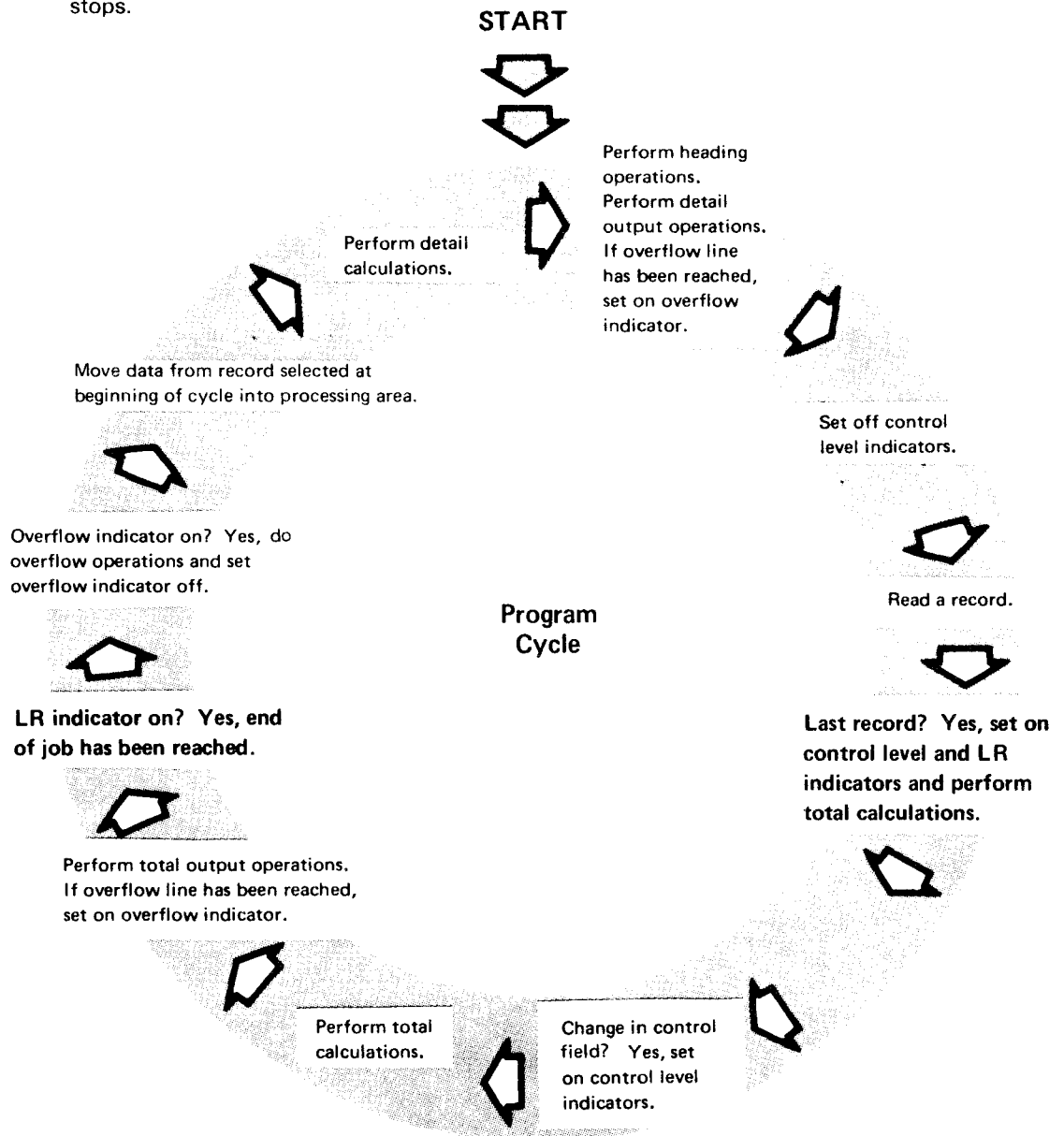


Figure 16. Program Cycle Operations for Last Record (LR) Indicator

JOB 4 (TRNREG): USING FIRST PAGE, OVERFLOW, AND LAST RECORD INDICATORS TO PRINT HEADINGS AND TOTALS

Job Definition

Print a weekly sales transaction report that lists daily transactions, total sales for the day, and total sales for the week. This report is similar to the one created in Job 3. The only difference is the addition of headings and final total.

The report title and column headings are printed on every page of the report. All items sold each day are listed. Item number, item description, quantity sold, unit price, and sales amount are included for every item. The date is printed for the first transaction in each group. After all transactions for a day are listed, the daily sales amount is printed. A final total of all daily sales is printed at the end of the report.

Job Requirements

Input: Sales transaction file consisting of 96-character records. Records are arranged in ascending order by date. The format of the input records is shown on this record layout form:

Transaction Date	Item Number	Item Description	Quantity	Unit Price																																												
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

Two decimal positions

Processing:

- Multiply quantity times unit price to find sales amount.
- Accumulate sales amount to find total item sales per day.
- Accumulate total daily sales to find total weekly sales.

JOB SPECIFICATIONS

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

IBM International Business Machines Corporation

GX21 9092 UM/050*
Printed in U.S.A.

Program	Transaction Register		Keying Instruction	Graphic	Card Electro Number	Page 02 of 2	Program Identification	TRNREG
Programmer	John Doe		Date	1/10/--				

File Description

The only new entry here is the OV indicator. Headings are to be printed at the top of every page. Overflow must be controlled in order to specify exactly where headings are to print. The overflow indicator is, therefore, assigned to the printer file to tell the compiler that it should not handle overflow automatically.

Line	Form Type	File Type	File Designation		Mode of Processing		Length of Key Field of Record Address Field	Record Address Field	Type of File Organization or Additional	Overflow Indicator	Key Field Starting Location	E.C.	K	Date	Entry	L	U	R	I	O	
			File Name	Sequence	File Format	Block Length															Record Length
02	F	TRAWS																			
03	F	REPORT								OV											
04	F																				PRINTER

RPG INPUT SPECIFICATIONS

IBM International Business Machines Corporation

GX21 9094 UM/050*
Printed in U.S.A.

Program	Transaction Register		Keying Instruction	Graphic	Card Electro Number	Page 02 of 2	Program Identification	TRNREG
Programmer	John Doe		Date	1/10/--				

Line	Form Type	Filename	Sequence	Record Identification Codes			Field Location		Field Name	Control Level (L1-L9)	Field Record Relation	Field Indicators									
				Position	Character	Position	Character	From				To	Plus	Minus	Zero or Blank						
01	I	TRANS	01																		
02	I							60	DATE	L1											
03	I							12	ITEMNO												
04	I							32	DESC												
05	I							37	QTY												
06	I							42	PRICE												

No new entries are made on the Input Specifications form.

RPG CALCULATION SPECIFICATIONS

IBM International Business Machines Corporation

GX21 9093 UM/050*
Printed in U.S.A.

Program	Transaction Register		Keying Instruction	Graphic	Card Electro Number	Page 03 of 3	Program Identification	TRNREG
Programmer	John Doe		Date	1/10/--				

Line	Form Type	Control Level (L0-L9)	Indicators			Factor 1	Operation	Factor 2	Result Field	
			And	And	Not				Name	Length
01	C					QTY	MULT	PRICE	EXTCST	72
02	C					EXTCST	ADD	DAYTOT	DAYTOT	82
03	C	L1				DAYTOT	ADD	WEEKTO	WEEKTO	92
04	C									
05	C									

Lines 01 and 02 are the same as for Job 3. Line 03, conditioned by L1, will be done only when a control break occurs. Once daily sales have been accumulated for a group (L1 is on), daily total can be added to the field WEEKTO used for accumulating weekly total. (Weekly total could also have been obtained by adding EXTCST to WEEKTO for every record read.)

Program **Transaction Register**
 Programmer **John Doe** Date **1/10/--**

Keying Instruction: _____
 Graphic Key: _____
 Card Electro Number: _____

Page **04** of **02** Program Identification: **TRNREG**

Line	Filename	Space		Skip			Output Indicators			Field Name	End Position in Output Record	Constant or Edit Word
		Before	After	Not	Not	Not	And	And	And			
01	REPORT	H	2	06			1P					
02		OR					OV					
03										58	'TRANSACTION REGISTER'	
04		H	1				1P					
05		OR					OV					
06										30	'TRANSACTION ITEM'	
07										52	'DESCRIPTION'	
08										78	'QUANTITY UNIT'	
09										89	'SALES'	
10		H	3				1P					
11		OR					OV					
12										29	'DATE NO'	
13										90	'PRICE AMOUNT'	
14		D	1				Ø1					
15							L1					
16										DATE	Y 18	

Program **Transaction Register**
 Programmer **John Doe** Date **1/10/--**

Keying Instruction: _____
 Graphic Key: _____
 Card Electro Number: _____

Page **05** of **02** Program Identification: **TRNREG**

Line	Filename	Space		Skip			Output Indicators			Field Name	End Position in Output Record	Constant or Edit Word
		Before	After	Not	Not	Not	And	And	And			
01												
02												
03												
04												
05												
06												
07												
08												
09												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												
33												
34												
35												
36												
37												
38												
39												
40												
41												
42												
43												
44												
45												
46												
47												
48												
49												
50												
51												
52												
53												
54												
55												
56												
57												
58												
59												
60												
61												
62												
63												
64												
65												
66												
67												
68												
69												
70												
71												
72												
73												
74												

Six different lines are to be printed, so six are described. The three heading lines should be printed either when 1P is on (first page) or when OV is on (overflow has occurred and forms have advanced). Thus, each heading is conditioned by 1P and OV used in an OR relationship. Notice the first heading line has the skip specification. This must be made before forms will advance.

The weekly total is printed only once on the report, after all records have been processed. It is conditioned by LR. A constant is specified for this total record so the words WEEKLY TOTAL will print as indicated on the printer spacing chart.

RECORD IDENTIFYING INDICATORS

In the jobs discussed so far, we have assumed that all records in the input file were alike. They did not necessarily contain the same information, but they had the same fields and the same kind of information in each field. They were of the same type.

However, in real jobs, often data files will not contain records of all the same type. Files often contain many types of records with different fields and different information. When using different record types in a job, you must have a way of telling the system what operations (calculations and output) you want done for each record read. Record identifying indicators are used for this.

Program Cycle Operations

Figure 17 shows program cycle operations associated with record identifying indicators. A record identifying indicator is set on right after a record is read and is set off before the next record is read.

Normally, record identification indicators condition detail calculations and detail output operations because detail operations are done for the record just read (the one associated with the record identifying indicator). On the other hand, total operations are not performed for any one record type; they are done after a certain number of records are processed.

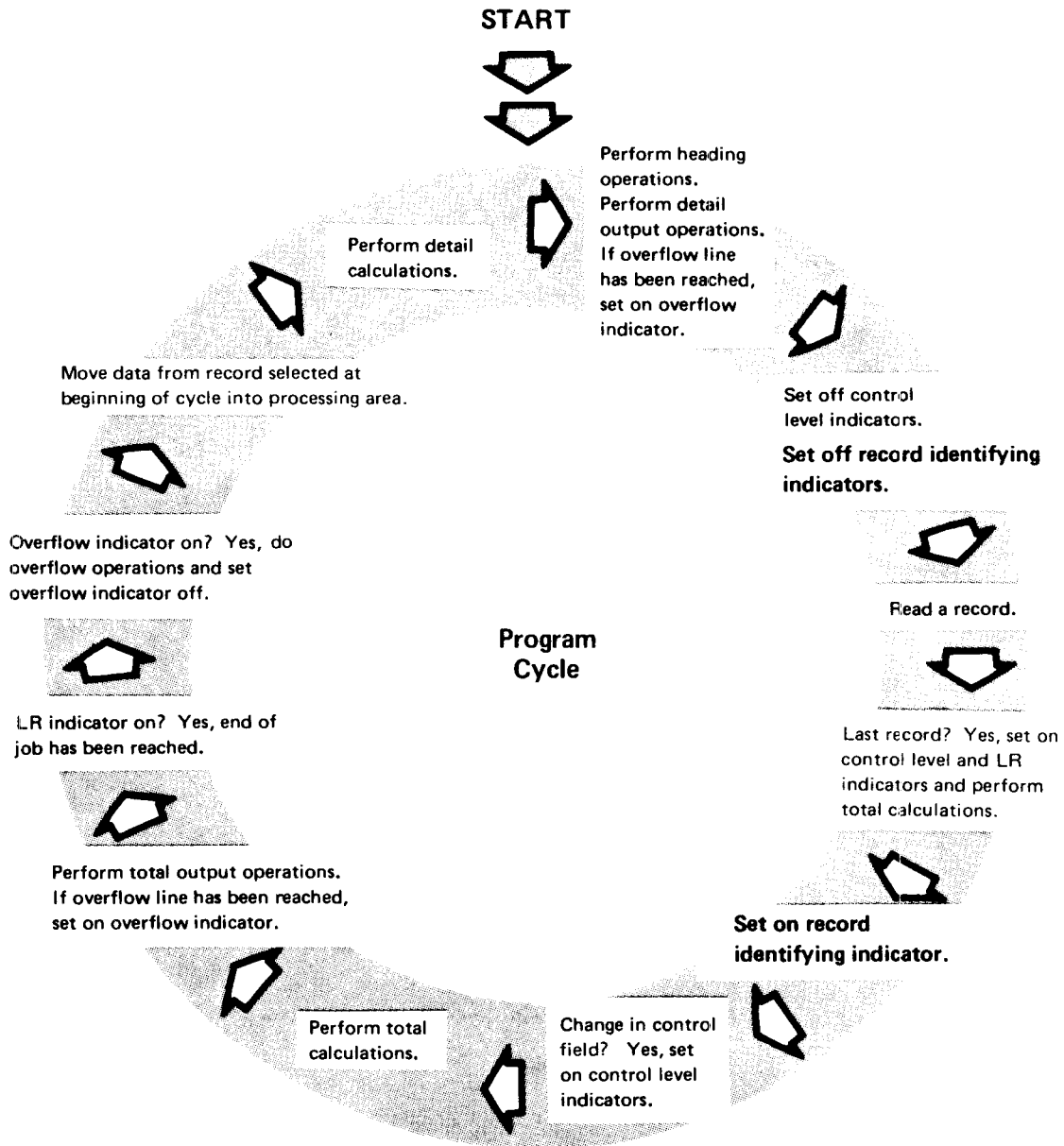


Figure 17. Program Cycle Operations for Record Identifying Indicators

RPG II Specifications

These are the RPG II specifications you must make when using different record types and record identifying indicators:

RPG INPUT SPECIFICATIONS

IBM International Business Machines Corporation

GX21-9094 UM-050*
Printed in U.S.A.

Program	Date	Keying Instruction	Graphic	Card Electro Number	Page 1 of 2	Program Identification
Programmer			Key			

Line	Type	Filename	Sequence	Record Identification Codes									Field Location		Field Name	Control Level (L, F, S)	Field Record Relation	Field Indicators	
				Position	Not (N)	Character	Position	Not (N)	Character	Position	Not (N)	Character	From	To				Plus	Minus or Blank
01	I			1			2			3									
02	I																		
03	I																		
04	I																		
05	I																		
06	I																		

RPG CALCULATION SPECIFICATIONS

IBM International Business Machines Corporation

GX21-9093 UM-050*
Printed in U.S.A.

Program	Date	Keying Instruction	Graphic	Card Electro Number	Page 1 of 2	Program Identification
Programmer			Key			

Line	Type	Control Level (L, F, S)	Indicators			Factor 1	Operation	Factor 2	Result Field		Comments
			And	And	Not				Name	Length	
01	C										
02	C										
03	C										
04	C										
05	C										

RPG OUTPUT SPECIFICATIONS

IBM International Business Machines Corporation

GX21-9090 UM-050*
Printed in U.S.A.

Program	Date	Keying Instruction	Graphic	Card Electro Number	Page 1 of 2	Program Identification
Programmer			Key			

Line	Type	Filename	Output Indicators			Field Name	Constant or Edit Word
			Space	Skip	Print		
01	O						
02	O						
03	O						

Assign a record identifying indicator (a 2-digit number from 01-99) to each record as ①, then use that same indicator to condition calculations at ② and detail output operations at ③, which must be done for that record only. Identify each record type by describing a code which identifies it at ④. If records must be in a certain order, indicate that sequence at ⑤.

If the records must be in a specific sequence, you must make entries in positions 15-18.

I		Filename													Number (1-99)	Record Identifying Indicator	Position	Not (N)	Character							
Line	Form Type	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0 1	I																									
0 2	I																									
0 3	I																									
0 4	I																									

- ① Positions 15-16 (Sequence) contain numeric entries (01-99). The sequence entries must be in numeric order.
- ② Position 17 (Number) tells how many records of this type to expect in the record-type group:
 - 1 = One and only one
 - N = One or more
- ③ Position 18 (Option) indicates whether the record type is required in the record-type group:
 - Blank = Record type is required
 - O = Record type is optional

Positions 15-16 should have an entry for every record type specified. If there is only one type per file the entry should be alphameric. If you fail to use a sequence entry, the compiler prints a message saying that no entry was made in positions 15-16.

Processing:

- Find total number of each item sold. To do this, perform the calculation $\text{ISSUE} + \text{TOTAL ISSUE} = \text{TOTAL ISSUE}$ for each issue record.
- Find total number of each item received. Perform the calculation $\text{RECEIPT} + \text{TOTAL RECEIPT} = \text{TOTAL RECEIPT}$ for each receipt record.
- When all transaction records for one item have been read, find new quantity onhand ($\text{ONHAND} + \text{TOTAL RECEIPT} - \text{TOTAL ISSUE} = \text{NEW ONHAND}$) and new quantity on order ($\text{ON ORDER} - \text{TOTAL RECEIPT} = \text{NEW ON ORDER}$).

Output: A printed stock status report:

STOCK STATUS REPORT				
ITEM NO	DESCRIPTION	QUANTITY ON HAND	QUANTITY ON ORDER	TRANSACTION QUANTITY
411116	B500 TWIN SOCKET BLUE ISSUE RECEIPT	458	500	50 500
		908**	0**	
411122	B506 SOCKET ADAPT BRN ISSUE ISSUE ISSUE	325	0	20 38 10
		257**	0**	
411173	C151C SIL SWITCH IVORY RECEIPT	50	150	150
		200**	0**	
411254	A210 PULL CORD GOLD ISSUE ISSUE	62	75	16 30
		16**	75**	

RPG CALCULATION SPECIFICATIONS

CX21 9093 UM 0507
Printed in U.S.A.

IBM
General Release Machine Corporation

Program Stock Status Keying Graphical Card Electro Number
 Programmer John Doe Date 1/10/-- Key

Page 03 of 75 76 77 78 79 80
 Program Identification STOKIST

C	Line	Form Tab	Indicators			Factor 1	Operation	Factor 2	Result Field		Decimal Positions	Resulting Indicators			Comments
			And	And	And				Name	Length		Arithmetic	Plus	Minus	
	01	C	20			ISSUES	ADD	TOTISU	TOTISU	50				FIND TOTAL SOLD	
	02	C	30			RECEET	ADD	TOTREC	TOTREC	50				FIND TOTAL RECD	
	03	C	L1			ONHAND	SUB	TOTISU	NEWONH	60				FIND NEW TOTAL	
	04	C	L1			NEWONH	ADD	TOTREC	NEWONH					ONHAND	
	05	C	L1			ONORD	SUB	TOTREC	NEWONO	60				FIND NEW ONORDR	
	06	C	L1			TOTISU	SUB	TOTISU	TOTISU					CLEAR TOTISU	
	07	C	L1			TOTREC	SUB	TOTREC	TOTREC					CLEAR TOTREC	

To update the quantity on hand and on order, total number issued (TOTISU) and total number received (TOTREC) for each item is needed. Quantity sold is found only in the issue record. Thus, the calculation to find TOTISU is done only when the issue record is read. Record identifying indicator 20 was assigned to the issue record. When 20 is on (an issue record has been read), we can calculate TOTISU. Thus, the operation (line 01) is conditioned by indicator 20. The operation to find TOTREC can be done only when a receipt record is read. The operation (line 02) is conditioned by 30, the record identifying indicator assigned to the receipt record.

Calculations to update quantity on hand and on order are total operations and can be done only after all transaction records for the item have been processed. They are conditioned by L1 which is set on when a new item number is read.

The operations on line 06 and 07 must be used to clear the total issues (TOTISU) and total receipts (TOTREC) fields after the quantity on hand and on order have been calculated.

Program Stock Status Keying Instruction _____ Graphic _____ Card Electro Number _____
 Programmer John Doe Date 1/10/-- Key _____

Line	Form Type	Filename	Type (M/D/T/E)				Space	Skip	Output Indicators			Field Name	End Position in Output Record	P/B/L/R	Constant or Edit Word
			M	D	T	E			1P	OV	10				
01	O	OUTPUT	H												
02	O		OR												
03	O														
04	O		H												
05	O		OR												
06	O														
07	O														
08	O														
09	O														
10	O		H												
11	O		OR												
12	O														
13	O														
14	O		D												
15	O														
16	O														
17	O														
18	O														
19	O		D												
20	O														
21	O														
01	O		O												
02	O														
03	O														
04	O		T												
05	O														
06	O														
07	O														
08	O														
09	O														
10	O														
11	O														
12	O														
13	O														
14	O														
15	O														
16	O														
17	O														
18	O														
19	O														
20	O														

Heading lines are to print on every page. They are conditioned by 1P and OV used in an OR relationship.

Not all detail lines are to print for each record. The first should be printed when the item master record is read. It is conditioned by 10, which is the record identifying indicator assigned to the item master record. The second detail line prints when an issue record is read, so it is conditioned by 20. The third detail line prints when a receipt record is read. It is conditioned by 30.

The printer spacing chart shows that the words ISSUE and RECEIPT are to print on the detail lines. These words are not fields so they are entered as constants on the appropriate detail lines. Note that asterisks indicating totals are also entered as constants on the total line.

RESULTING INDICATORS

Sometimes your decision to do a certain operation is based on the result of a previous operation. Resulting indicators allow you to specify which operations you want done and the conditions under which the operations are to be done. Resulting indicators can be used to determine such things as:

- Whether a result is larger, smaller, or equal to a predetermined number.
- Whether a certain result is plus, minus, or zero.

Program Cycle Operations

Figure 19 shows the operations in the program cycle associated with resulting indicators. Resulting indicators are set when the associated calculation operation is performed. This means that resulting indicators can be set either at detail or at total calculation time.

Resulting indicators are not set off automatically. They change their setting only at the time a calculation is performed or when they are set off intentionally. For example, if a resulting indicator is set on by a detail calculation, it retains this setting until the next time it is used as a resulting indicator.

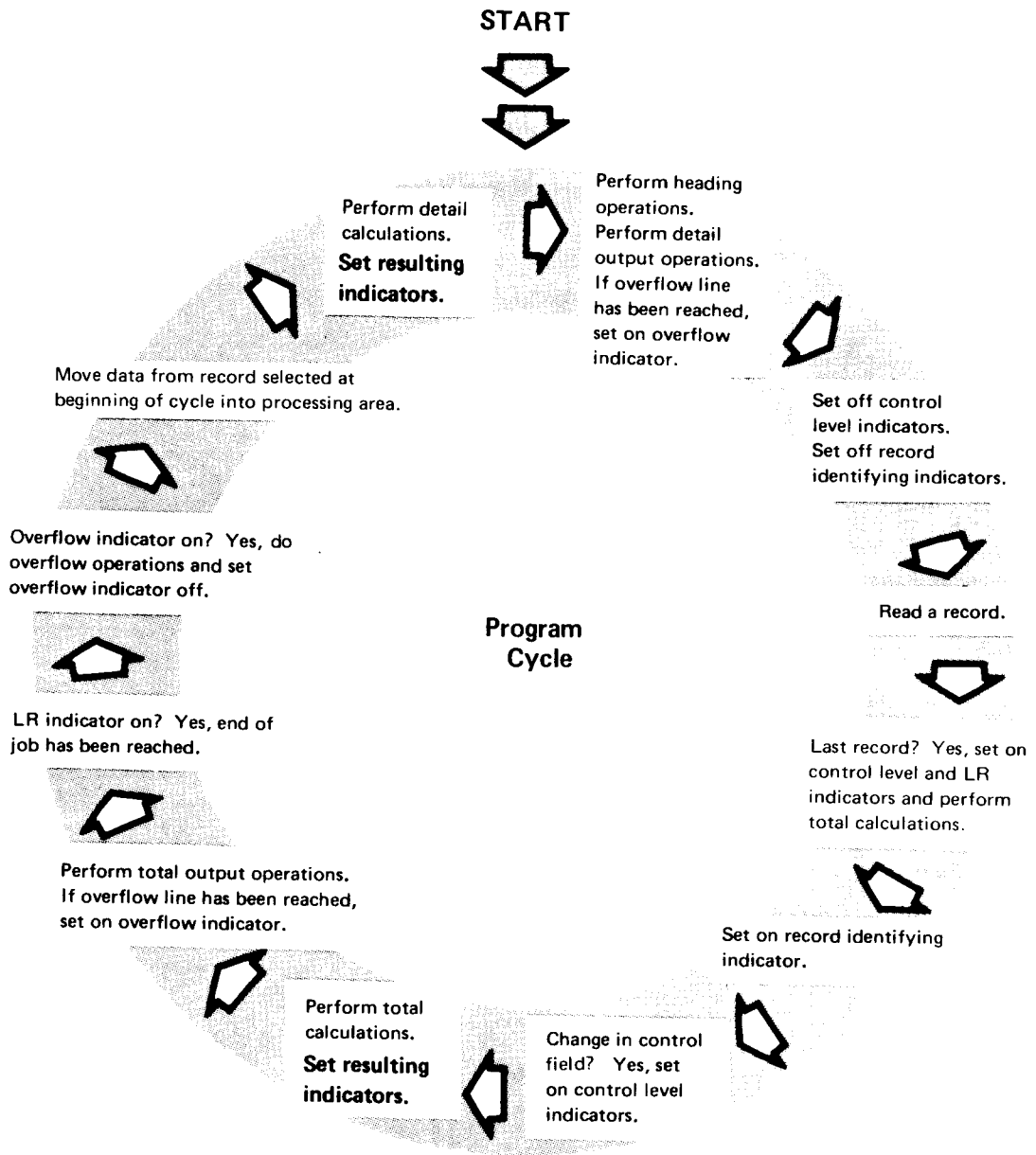


Figure 19. Program Cycle Operations for Resulting Indicators

When testing for more than one condition, you can use the same or different indicators in these positions. If you intend to do different operations for each of the three conditions, enter a different resulting indicator to test for each condition on the Calculation Specifications form:

IBM RPG CALCULATION SPECIFICATIONS

G221 9093 (10 65)
Printed in U.S.A.

Program: _____ Page: 1 of 2
 Programmer: _____ Date: _____ Key: _____ Graphic: _____ Card Electro Number: _____
 Extraction: _____ Key: _____ Program Identification: 75 76 77 78 79 80

Line	C	Indicators		Factor 1	Operation	Factor 2	Result Field		Resulting Indicators			Comments
		And	And				Name	Length	Arithmetic	Plus	Minus	
01	C			MAX	COMP	TOTAL			11	22	33	
02	C	11		TOTAL	MULT	.05	COMM	72	H			
03	C	22		TOTAL	MULT	.06	COMM		H			
04	C	33		TOTAL	MULT	.07	COMM		H			

In this operation, the contents of the field MAX is compared to the contents of the field TOTAL. If MAX is greater than TOTAL, indicator 11 is set on and the operation on line 02 is done. If MAX is less than TOTAL, indicator 22 is set on and the operation on line 03 is done. If MAX is equal to TOTAL, indicator 33 is set on and the operation on line 04 is done. Only the resulting indicator for the condition that is true is set on. The other resulting indicators are set off.

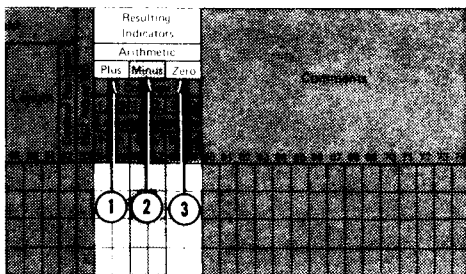
We explained previously that constants can be used in arithmetic calculation operations, but they must be numeric constants. In a COMP operation, however, constants can be either alphameric or numeric. Rules for using alphameric constants as Factor 1 or Factor 2 are a little different from those for using numeric constants:

Rules for Numeric Constants	Rules for Alphameric Constants
<ul style="list-style-type: none"> • A numeric constant can be any combination of digits 0-9. Decimal points and signs can also be included. • The maximum length of a numeric constant is 10 characters, including sign and decimal point. • Numeric constants must not be enclosed in apostrophes ('). 	<ul style="list-style-type: none"> • An alphameric constant can be any combination of characters (letters, digits, and special characters). Blanks are also valid. • The maximum length of an alphameric constant is 8 characters. • Alphameric constants must be enclosed in apostrophes (').

When you use the COMP operation code, remember to always compare two numeric fields or constants or two alphameric fields or constants. You cannot compare a numeric field or constant to an alphameric field or constant.

Using an Arithmetic Operation

You can test the results of an arithmetic operation (ADD, SUB, MULT, DIV) for plus, minus, or zero by entering resulting indicators in the appropriate positions on the Calculation Specifications form:



- ① A resulting indicator entered in positions 54-55 tells the system to determine if the result field is positive (plus).
- ② A resulting indicator entered in positions 56-57 tells the system to determine if the result field is negative (minus).
- ③ A resulting indicator entered in positions 58-59 tells the system to determine if the result field is zero.

The tests you indicate are performed each time the operation is executed. However, the assigned indicator is set on only if the field satisfies the condition tested. If you entered indicator 99 in positions 54-55 to test the result field for plus, indicator 99 would be set on only if the result field were plus. If the condition is not met, the indicator is set off.

Again, as with the COMP operation, you can test for one, two, or all three conditions at the same time. When testing for more than one condition, you can use the same or different indicators in these positions. If you intend to do different operations for each of the three conditions, enter a different resulting indicator to test for each condition:

IBM International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

GX21 9093 UM 0507
Printed in U.S.A.

Program	Keying Instruction	Graphic	Card Electro Number
Programmer	Date	Key	

Page of Program Identification

Line	Form Type	Control Level (L0/L1/L2)	Indicators			Factor 1	Operation	Factor 2	Result Field		Decimal Positions	Resulting Indicators	Comments
			Pos	Zero	Neg				Name	Length			
01	C					FIELDA	ADD	FIELDB	FIELDC	60			
02	C		10			FIELDC	SUB	50	FIELDC			102549	
03	C		25			FIELDC	SUB	100	FIELDC				
04	C		49			FIELDC	ADD	10	FIELDC				
05	C												
06	C												
07	C												
08	C												
09	C												
10	C												
11	C												
12	C												
13	C												
14	C												
15	C												
16	C												
17	C												
18	C												

FIELDC is tested for all three conditions. If the field is positive, indicator 10 is set on and the operation on line 02 is performed. If the field is negative, indicator 25 is set on and the operation on line 03 is done. If the field is zero, indicator 49 is set on and the operation on line 04 is done.

If you want to do the same operations when the result field meets either one of two conditions (plus or zero, minus or zero), you could use the same indicator to test for both:

IBM International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

GX21-9063 UM-9597
Printed in U.S.A.

Program	Date	Keying Instruction	Graphic	Card Electro Number	Page 1 2 of	Program Identifier
Programmer						

C	Line	Form Type	Control Level (L0 L9 LP SP AN/OP)	Indicators						Factor 1	Operation	Factor 2	Result Field			Resulting Indicators			Comments			
				Not	And	Not	And	Not	Not				Name	Length	Decimal Position	High	Low	Equal		Authentic	Plus	Minus
	01	C							FIELD A	ADD	FIELD B	FIELD C	60	10	20	10						
	02	C		10					FIELD C	SUB	50	FIELD C										
	03	C		20					FIELD C	ADD	100	FIELD C										

FIELD C is tested for three conditions, but only two indicators are used. If FIELD C is either plus or zero, indicator 10 is turned on and the operation on line 02 is performed. However, if FIELD C is minus, indicator 20 is set on and the operation on line 03 is performed.

JOB 6 (STOKST): USING RESULTING INDICATORS TO TEST CONTENTS OF RESULT FIELDS

Job Definition

Print a stock status report similar to the one in Job 5. The only difference is the addition of maximum and minimum balances. Item master records usually include the maximum and minimum on-hand quantity for all items. These figures are kept so that checks can be made, whenever the inventory is updated, to determine if quantity on hand is within the limits set.

The first line for each item in the report shows standard descriptive data for the item: item number, item description, quantity on hand, quantity on order, maximum and minimum balances. Subsequent lines give the detail on current transactions involving the item. Quantities remaining on hand and on order are calculated for each item and printed after all transactions for the item are listed. Whenever shipments reduce stock on hand below the predetermined minimum balance or whenever receipts push the quantity on hand above the predetermined maximum, an exception condition is noted on the report.

Job Requirements

Input: An inventory file consisting of three different record types. Formats of the three record types are:

Name ITEM MASTER RECORD

CODE=M	ITEM NUMBER	DESCRIPTION	UNIT COST	QUANTITY ON HAND	QUANTITY ON ORDER	MAX BAL	MIN BAL																																																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59

Two decimal positions

Name ISSUE RECORD

CODE = I	Item Number							Quantity Sold																																																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60

Name RECEIPT RECORD

CODE = R	Item Number							Quantity Received																																																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60

The file is organized in ascending order by item number. For each item, one master record is required. Issue and receipt records are optional. When present, however, there may be any number of each. Records for each item are in this order:

1. Item master
2. Issue(s)
3. Receipt(s)

Processing:

- Find total number of each item sold. To do this, perform the calculation $\text{ISSUE} + \text{TOTAL ISSUE} = \text{TOTAL ISSUE}$ for each issue record.
- Find total number of each item received. To do this, perform the calculation $\text{RECEIPT} + \text{TOTAL RECEIPT} = \text{TOTAL RECEIPT}$ for each receipt record.
- When all transaction records for one item have been read, find new quantity on hand ($\text{ON HAND} + \text{TOTAL RECEIPT} - \text{TOTAL ISSUE} = \text{NEW ON HAND}$) and new quantity on order ($\text{ON ORDER} - \text{TOTAL RECEIPT} = \text{NEW ON ORDER}$).
- Compare the new quantity on hand to maximum and minimum balances to determine if an exception condition should be noted on the report.

Program: **Stock Status**
 Programmer: **John Doe** Date: **1/10/--**
 Keying Instruction: _____ Graphic: _____ Card Electro Number: _____
 Key: _____

Page **03** of **12**
 Program Identification: **SITOKST**

Line	Code	Factor 1	Operation	Factor 2	Result Field		Resulting Indicators			Comments
					Name	Length	Arithmetic	Plus	Minus	
01	C	ISSUES	ADD	TOTISU	TOTISU	50				FIND TOTAL SOLD
02	C	RECEPT	ADD	TOTREC	TOTREC	50				FIND TOTAL REC.
03	C	ONHAND	SUB	TOTISU	NEWONH	60				FIND NEW TOTAL
04	C	ONHAND	ADD	TOTREC	NEWONH					ONHAND
05	C	ONORD	SUB	TOTREC	NEWONO	60				FIND NEW ONORDR
06	C	NEWONH	COMP	MAX			99			IS ONHAND > MAX
07	C	NEWONH	COMP	MIN				88		IS ONHAND < MIN
08	C	TOTISU	SUB	TOTISU	TOTISU					CLEAR TOTISU
09	C	TOTREC	SUB	TOTREC	TOTREC					CLEAR TOTREC

Calculations on lines 01-05, 08, and 09 are needed to update quantity on hand and quantity on order. See Job 5 for an explanation of these entries. After new quantity on hand (NEWONH) has been calculated, it is compared to MAX to see if it exceeds the maximum limits set (line 06). Indicator 99 in positions 54-55 specifies a test to determine whether Factor 1 (NEWONH) is greater than Factor 2 (MAX). If NEWONH is greater, indicator 99 is set on. On line 07, NEWONH is compared to MIN to see if quantity on hand is less than the minimum set. If it is, indicator 88 is set on.

FIELD INDICATORS

Field indicators, like resulting indicators, are used to test the contents of a field and to condition operations based on the results of the test.

Program Cycle Operations

Figure 20 shows the operations in the program cycle associated with field indicators. Note that input fields are tested and field indicators are set to reflect the result of the test at the time data is moved into the processing area. Field indicators are not set off at the end of the program cycle. If a field indicator is set on when data is moved into the processing area in the first cycle, it is not reset until the appropriate field is moved into the processing area in a subsequent cycle.

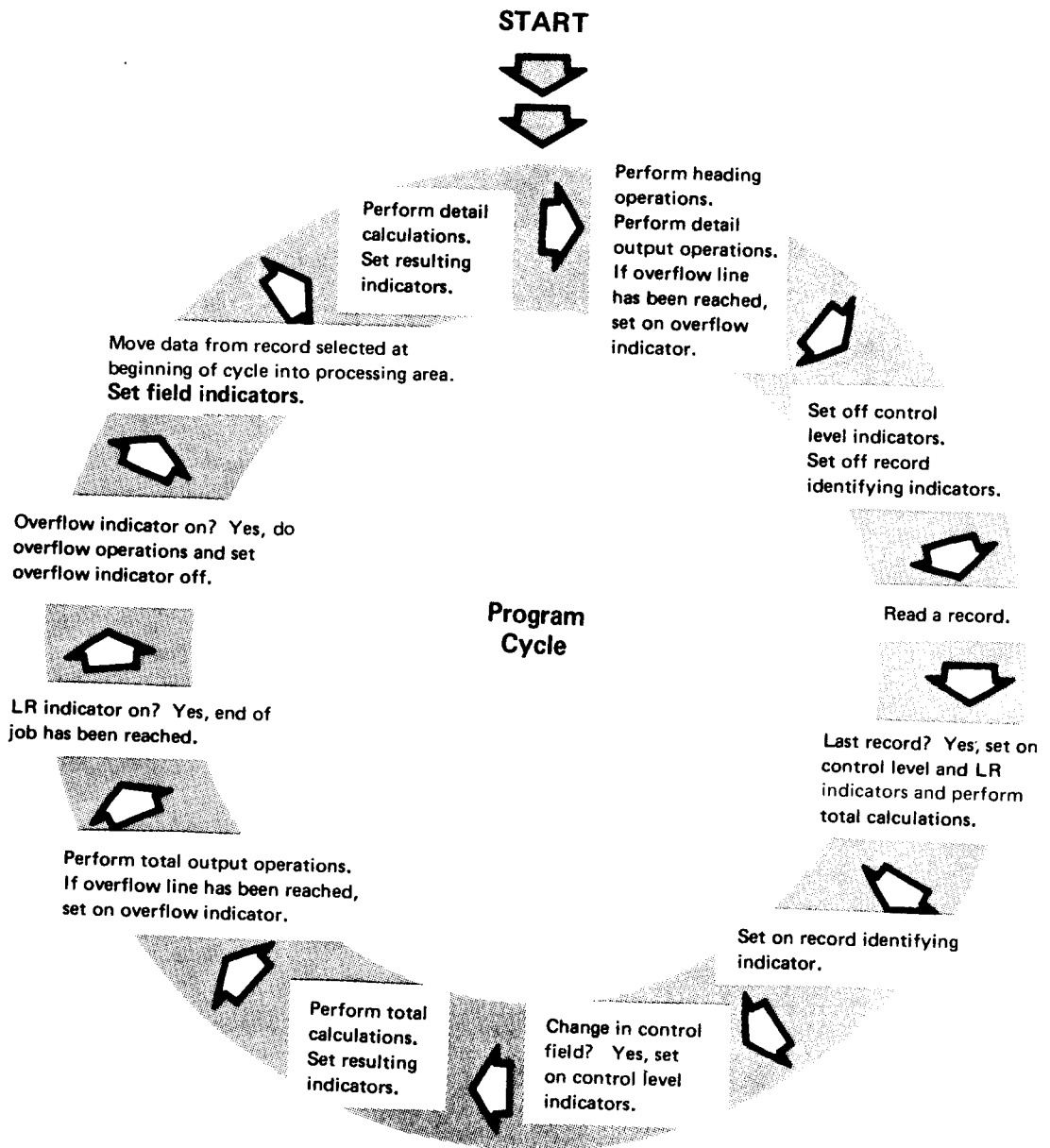
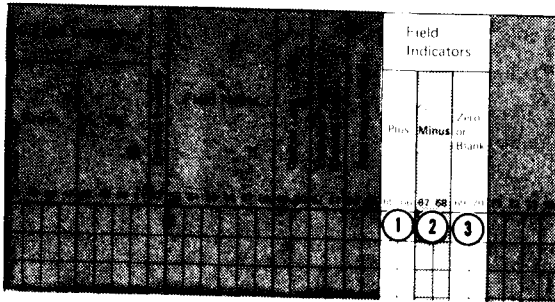


Figure 20. Program Cycle Operations for Field Indicators

You can enter any one of the indicators 01-99 in positions 65-70 on the Input Specifications form to test an input field. You may assign indicators to test for three possible conditions:



- ① A field indicator assigned in positions 65-66 tells the system to determine if a numeric input field is positive (plus).
- ② A field indicator assigned in positions 67-68 tells the system to determine if a numeric input field is negative (minus).
- ③ A field indicator assigned in positions 69-70 tells the system to determine if an alpha-numeric input field is blank or a numeric field is zero.

JOB 7 (AGETB): USING FIELD INDICATORS TO TEST CONTENTS OF INPUT FIELDS

Job Definition

Create an aged-trial balance report that lists:

- Name and customer number of all charge customers who have payments due.
- Amount due.
- Overdue balances.

The customer master file, which contains records for all customers regardless of their balance, is used as the input file. The report is to show only those customers with payments due. Thus, information from customer records that contain a zero or credit balance should not be printed.

Job Requirements

Input: A customer master file consisting of one record type:

CODE = M	Customer Number	Customer Name	Last Payment Date	Credit Limit	Current Charges	30 Days Overdue	60 Days Overdue	90 Days Overdue
	1-6	7-19	31-39	40-41	42-48	49-57	58-62	63-71

Processing: Check input balance due field for zero or credit balance (use field indicators).

Output: A printed aged-trial balance report:

AGED TRIAL BALANCE							
CUSTOMER NUMBER	CUSTOMER NAME	LAST PAY DATE	CREDIT LIMIT	CURRENT CHARGES	OVERDUE ACCOUNTS		
					30 DAYS	60 DAYS	90 DAYS
10867	ALLEN E. CO.	2/16/ --	15,000.00	6,919.77			
16535	ANDERSON AUTO SUPPLY	1/28/ --	2,500.00	1,665.49	375.58		
17849	ANDREWS AND SUNS INC	2/05/ --	750.00				
18978	AKGONAUT ENGINEERING	12/27/ --	2,000.00	2,111.30		145.54	
24743	BERKLEY PAPER CO	2/21/ --	6,300.00	1,185.50	611.54	312.13	93.44
25271	BEST DISTRIBUTION CO	10/06/ --	1,000.00	3.25	2,652.45	1,400.75	51.00
							762.19

This printer spacing chart shows how the report is formatted:

AGED TRIAL BALANCE											
CUSTOMER NUMBER	CUSTOMER NAME	LAST PAY DATE	CREDIT LIMIT	CURRENT CHARGES	OVERDUE ACCOUNTS						
					30 DAYS	60 DAYS	90 DAYS				
XXXX	XXXXXXXXXXXXXXXXXXXX	XX/XX/XX	XX,XXX.XX	XX,XXX.XX	XX,XXX.XX	XX,XXX.XX	XX,XXX.XX				
Single space detail lines.											

If your calculation or output operation must be conditioned by more than three indicators, additional indicators can be specified on the next line if AND is entered in positions 14, 15, and 16 of the Output Specifications form or AN is entered in positions 7-8 of the Calculation Specifications form:

RPG OUTPUT SPECIFI

IBM International Business Machines Corporation

Program										Keying Instruction										Graphic																					
Programmer										Date										Key																					
Line	Form Type	Filename	Type (H/D/T/E)			Space		Skip		Output Indicators			Field Name										End Position in Output Record																		
			D	R	A	Before	Alter	Before	Alter	And	And	And																													
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
01	O																																								
02	O																																								
03	O																																								
04	O																																								

Five indicators used in an AND relationship condition this detail record. Three indicators are specified on one line; the remaining are specified on the following line with the word AND in positions 14-16. Indicators 10, 11, 12, 15, and 16 must all be on before the detail line will be printed.

RPG CALCULATI

IBM International Business Machines Corporation

Program										Keying Instruction										Graphic																				
Programmer										Date										Key																				
Line	Form Type	Control Level (L0, L9, LR, SR, AN/OR)	Indicators			Factor 1	Operation	Factor																																
			And	And	And																																			
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40			
01	C																																							
02	C																																							
03	C																																							
04	C																																							

This calculation operation is done only if indicators 25, 30, 31, and 90 are on and 91 is not on. Note that the operation is specified on the AN line.

Chapter 3. The Programmer's Job

Your responsibilities as a programmer include:

- Determining the job requirements
- Determining what RPG II specifications and program cycle operations are needed for the job
- Writing the specifications
- Documenting the program
- Preparing your source program for compilation
- Compiling the source program
- Testing the program

DETERMINING THE JOB REQUIREMENTS

The requirements for a job are generally described in terms of the input provided and the output required. The following paragraphs and illustrations describe the job requirements.

An invoice is to be prepared like that shown below:

INVOICE				
ACCOUNT NUMBER 09621				
NAME	SMITH MANUFACTURING			
ADDRESS	13620 9TH ST NE BERNALILLO NEW MEXICO 56120			
SHIPPING INSTRUCTIONS BY AIR				
ITEM NUMBER	DESCRIPTION	QUANTITY	UNIT PRICE	AMOUNT
439167	SHEARS	100	27.56	2,756.00
629408	GASKET CORK	3000	1.15	3,450.00
102139	SPRIDGET WHITE	50	750.00	37,500.00
INVOICE TOTAL				212,157.92

The input file contains two types of records: name/address records for all customers who made purchases on credit during the month and transaction records for each item purchased by the customers during the month. The name/address and transaction records look like this:

Name NAME/ADDRESS RECORD

CODE = N	ACCOUNT NUMBER	NAME	ADDRESS LINE 1	ADDRESS
	1 2 3 4 5 6	7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60		

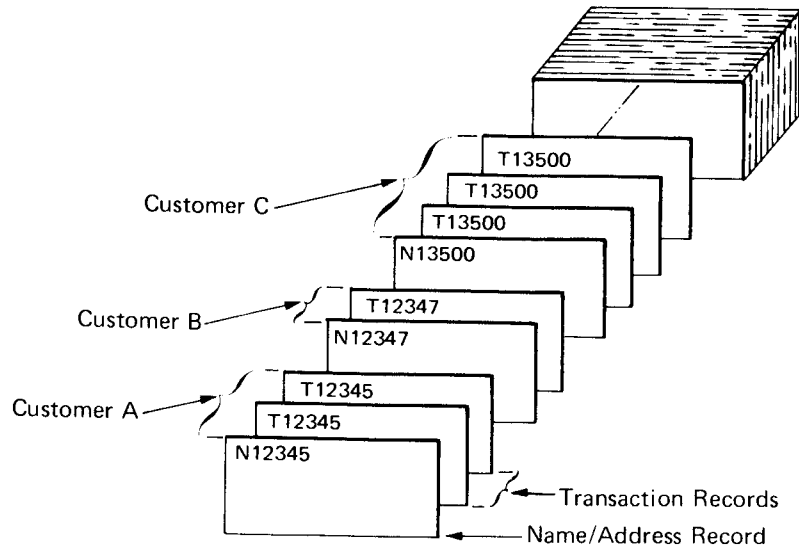
LINE 2	ADDRESS LINE 3
61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96	

Shipping code (either 1, 2, or 3)

Name TRANSACTION RECORD

CODE = T	ACCOUNT NUMBER	ITEM NUMBER	DESCRIPTION	QUANTITY	UNIT PRICE
	1 2 3 4 5 6	7 8 9 10 11 12 13 14	15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34	35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60

The input file is organized so that all transaction records for a customer follow the customer's name/address record. Each customer has one name/address record, but might have one or more transaction records.



Your first step is to analyze the problem and decide what processing must be done to get the desired results. Always keep in mind how things are done using RPG II. In your analysis of the job, you would probably think of these points:

- Information for the first part of the invoice is taken from the name/address record. Information for the second part (list of transactions) is taken from transaction records.

- In order to print shipping instructions, the shipping code recorded in the record must be determined:

1 = By truck

2 = By rail

3 = By air

- AMOUNT and INVTOT (invoice total) must be calculated because this information is not in the input records. These calculations must be done for all transaction records:

$QTY \times PRICE = AMOUNT$

$AMOUNT + INVTOT = INVTOT$

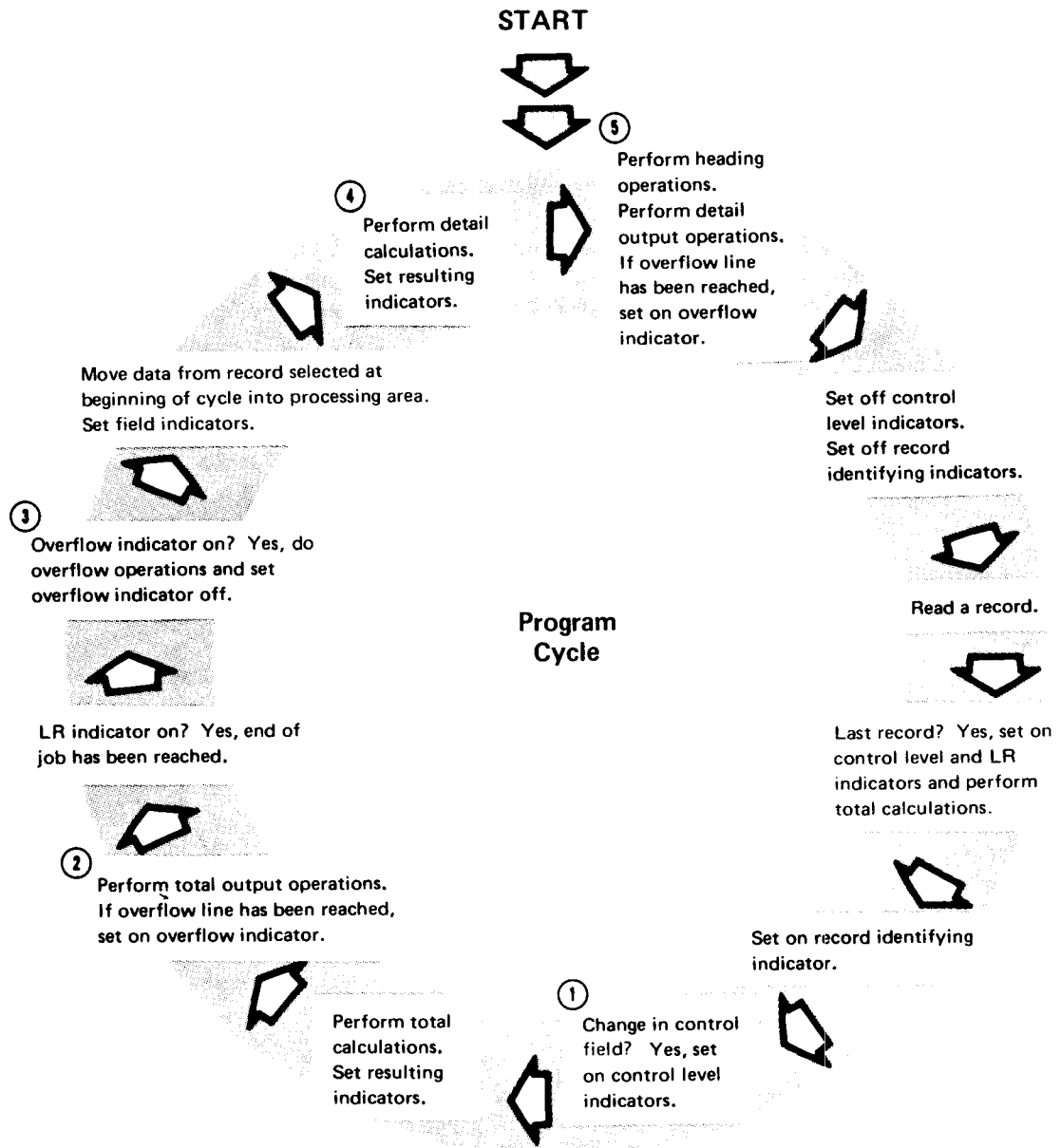
- INVTOT should be printed only after all transaction records for one account have been processed.
- The invoice for each customer must be on a separate page. This means that forms must advance each time a new customer name/address record is found. It is possible that one customer has purchased so many items that they cannot be listed on one page. In this case, forms should advance when the end of a page is reached. When an invoice includes more than one page, headings should be printed on all pages.

DETERMINE RPG II SPECIFICATIONS NEEDED FOR THE JOB

After you have carefully analyzed the job, determine what RPG II specifications and program cycle operations you need. For example, consider the following:

- Different record types are used. This means that record identifying indicators must be specified to tell what to do for each record.
- The shipping code must be determined. One way to do this is to compare the shipping code to 2. Through the use of resulting indicators, you can determine if the code is less than 2 (1), greater than 2 (3), or equal to 2.
- INVAMT is printed only after all transaction records for one account have been processed. This is a total operation, done only after a group of records has been processed. Therefore, control fields and control level indicators must be used to do a total operation. The account number field can be used as the control field.
- Forms should advance each time a different name/address record is encountered or whenever overflow occurs. Thus heading lines must be conditioned by a record identifying indicator and the OV indicator.

If the indicators and steps just listed are used, the RPG II program cycle would include the steps shown in Figure 21.



- ① Did Account Number change?
- ② Print total only after all transaction records for one customer have been processed.
- ③ Print all heading lines if overflow occurs.
- ④ The operation to find shipping instructions can be done only for name/address record. The operations to find AMOUNT and INVTOT can be done only for transaction records.
- ⑤ Headings for the invoice can be printed only when name/address record is read and detail lines only when transaction records are read.

Figure 21. Program Cycle Operations for Sample Job

DOCUMENT THE PROGRAM

An important part of every programmer's job is to explain his program. This documentation provides information for people who will run the program and for programmers who may later need to alter or update it. Documentation is also useful to you. It is not always easy to remember what every program you wrote does. Reading documentation is a much easier way to recall the program than figuring out each instruction.

Documentation consists of:

1. Telling generally what the program does.
2. Describing input and output. (Record layout forms and printer spacing charts are an excellent means of describing input and output.) File names and field names should be meaningful.
3. Explaining the coding.
4. Telling the operator how to run the program, what to do if the system stops because of an error, and what to do when the job is completed.

All documentation cannot be done at the time you write specifications. However, when writing your specifications, you can also write an explanation for a line or lines of coding on the specifications forms. You have probably noticed positions labeled Comments on the specifications sheets. Here is where you write an explanation for your coding.

In addition to using the comment positions on the coding forms, you can use comment lines. A comment line is indicated by an * (asterisk) in position 7 of the coding form:

RPG INPUT SPECIFICATIONS

GX21 9094 UM/050*
Printed in U.S.A.

Program		Keying Instruction		Graphic		Card Electro Number	
Programmer		Date		Key		Page 1 of 2	

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Record Identification Codes										Field Location		Field Name	Field Indicators				
			Position	Character	Position	Character	Position	Character	Position	Character	From	To	Plus	Minus		Zero or Blank				
01	I	*A COMMENT LINE IS INDICATED BY AN * IN COLUMN 7. A COMMENT LINE CAN																		
02	I	*INCLUDE ANY INFORMATION YOU WANT.																		
03	I	* * * * *																		
04	I	* * * * *																		
05	I	* * * * *																		
06	I	* * * * *																		
07	I	* * * * *																		
08	I	* * * * *																		
09	I	* * * * *																		

Comment lines can be used anywhere on any specifications form. There is no limit to the number you can use. The RPG II compiler does not regard comments and comment lines as part of the program. Therefore, the compiler does not translate the comments into instructions; however, the lines are printed as part of the source program listing.

PREPARE FOR COMPILATION

After completing your source program, you must prepare it for compilation.

Specifications Form Order

Your specifications forms must be in this order:

1. Control and File Description Specifications form
2. Input Specifications form
3. Calculation Specifications form
4. Output Specifications form

Number the forms in positions 1 and 2. At this time, you might also check to see that the top part of each form is completely filled in.

If you are planning to give these specifications to someone to key, it is a good idea to fill in the box labeled Keying Instructions:

IBM International Business Machines Corporation										RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS									
Program					Keying Instruction					Graphic					Card Electro Number				
Programmer					Date					Punch									

You indicate in this box the graphic symbols you are using and their meaning. Some printed letters and numbers are easily confused. For example, it is sometimes difficult to differentiate between the number 0 and the letter O and the number 2 and the letter Z. You may, therefore, devise a graphic symbol that you use for certain letters. Some people use Ø for zero, Z̄ for the letter Z. Explain your symbols so that the operator will know what to key when the symbol appears on the coding forms.

Notice that no line numbers have been entered in positions 3-5 of the specification lines located below line 20. You can place numbers in these positions to tell where the missing specification belongs:

RPG OUTPUT SPECIFICATIONS

GX21-9090 UM/050
Printed in U.S.A.

IBM International Business Machines Corporation

Program	Keying Instruction	Graphic	Card Electro Number
Programmer	Date	Key	

Page 1 of 2
Program Identification 75 76 77 78 79 80

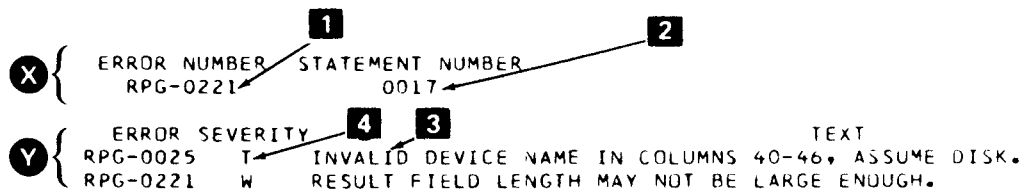
Line	Form Type	Filename	Type (INDU/TYPE)		Space	Skip	Output Indicators			Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word
			O	N			And	And	Not				
01	O	INVOICE	H	304			01						
02	O		OR				OV						
03	O												
04	O		H	3			01				45	'INVOICE'	
05	O		OR				OV						
06	O		H	2			01						
07	O		OR				OV						
08	O												
09	O												
10	O		H	2			01				7	'NAME'	
11	O		OR				OV				38		
12	O												
13	O												
14	O												
15	O												
16	O												
17	O												
18	O												
19	O												
20	O												
051	O										27	'ACCOUNT NUMBER'	
052	O										24		

In this example, the programmer did not specify information to be included on the second heading line. On the lines following line 20, he entered the missing specifications. Notice that positions 3-5 tell where the specifications belong. The line numbers 051 and 052 indicate that the specifications belong between lines 05 and 06.

If your specifications are being keypunched, the out-of-order cards must be inserted in the appropriate place. If the source program is being entered directly into the system through a keyboard, the missing specifications will have to be inserted in the appropriate place when the specifications are keyed.

If the compiler finds any errors in your source specifications, it will print diagnostic messages telling you what errors were made. You will find that different types of messages are printed: warning, terminal, or informative. A warning message is an indication that something may be wrong. If you check the questioned specification and find that it is all right for your program, you need not make changes. If you get a terminal message, however, something is wrong with your coding. You must fix the specification and recompile the program before the compiler will actually translate your specifications.

The diagnostic message section of the program listing contains two basic parts: a list of messages (X) and an explanation of each message (Y).



Each error message in the list is identified by a 3-digit number (1). Next to the message number is either a statement number identifying the specification in which the error appears or a field name or constant associated with the error (2). Following the list of messages is an explanation of each error (3) and an indication of the severity of each error (W = warning; T = terminal) (4).

The sample shown above shows diagnostic messages printed for the invoice job. Note that message 221 is a warning. A warning is an indication that something may be wrong. If you check the specification noted and find that it will work for the job, you need not change it.

Checking the message in the listing, you would find that the warning points to the AMOUNT field in statement 0017:

```
0017 0018 C 20 QTY MULT UPRICE AMOUNT 52 FIND ITEM TOTALINVOCE
```

The AMOUNT field is specified as five characters with two decimal positions. In checking your specifications, you will find that the amount should have been 7 digits with 2 decimals, and it was incorrectly keyed in the source data.

Message 0025 is a terminal error. This error number appears below the statement at which the error occurred. Checking the listing where it appears at statement 1, the device entry DISK, was miskeyed as DSIK. You must correct this error. (Check the RPG II reference manual for your system for the correct entries for the Device positions.)

TEST THE PROGRAM

It is good practice to test your program before using it for an actual job. To do this, make up test data representing all possible situations that could arise during an actual job. Run your program using that data to see if your program will really handle the situations you think it does. If you get the wrong results when testing, you know your program is not doing what you thought it would. You can usually find your errors by using actual input data and doing the operations specified yourself, step-by-step, in the order the system would do them. When doing this, you have to follow closely your specifications and the program cycle operations taken by your program. After you test your program and the results show it can handle all situations, your job is complete.

address: A number identifying a location in storage.

alphabetic: In general usage, any of the letters A-Z. In RPG II programming, any of the letters A-Z and special characters @, #, and \$.

alphameric: Any alphabetic or numeric characters.

alphameric constants: Any of the characters in the data character set.

alphameric field: A field that contains any of the characters in the data character set.

AND relationship: The specifying of conditioning indicators such that the operation conditioned will be performed only when all conditions are met.

arithmetic/logic unit: An area inside the processing unit where calculations are performed.

arithmetic operation: An operation such as addition, subtraction, multiplication, and division performed in the processing unit.

ascending order: The arrangement of records in a file from low to high, based on the contents of a specified field in each record.

blank after: Changing the contents of a field so that it contains only zeros or blanks after that field has been written to the output record.

Calculation Specifications form: An RPG II coding form which specifies the type and order of calculations to be performed on the input data.

card: In data processing, a card containing combinations of holes representing data to a system.

card file: A group of related punched-card records.

card layout form: A chart for planning the design and format of cards.

card punch: A device that records information on a card in the form of combinations of holes representing characters.

card reader: A device that electronically senses information on punched cards and transfers that information to the processing unit.

character: Any individual data item that can be represented in printed form; that is, a letter, a digit, or a special character.

coding: Making entries on RPG II specification forms.

comments: Words or statements in a program that serve as documentation rather than as instructions to the compiler.

compile: Translate a source program (such as RPG II specifications) into an object program (machine language program) using the compiler.

compiler: A program that translates a source program into a machine language program.

conditioning: Using indicators to control when calculations or output operations are done.

constant: A data item that does not change during execution of a program. This item represents itself and is actually used in processing rather than being a field name representing the data. For example, COST is a name representing a field containing data that changes, whereas the constant 100 is actual data used that does not change.

control break: A change in the contents of a control field.

Control and File Description Specifications form: An RPG II coding form that gives, for a particular job, information needed for control of the system and a description of the files used.

control field: One or more fields that are compared from record to record to determine when certain operations should be performed.

control group: A set of records all having the same control field information.

control level indicator: An indicator used to specify certain fields as control fields and to indicate which operations to perform at total time.

control unit: An area inside the processing unit that determines from instructions what has to be done. It directs other units or devices to perform the required functions.

data: A collection of facts, numbers, letters, and symbols that can be processed or produced by a system.

data character set: All of the 256 EBCDIC characters.

descending order: The arrangement of records in a file from high to low, based on the contents of a specific field in each record.

detail record: An output record produced during the detail output operation of the RPG II program cycle.

detail time: An operation in the RPG II program cycle in which calculation and output operations are performed for each record read.

diagnostic message: An output message that identifies RPG II specification errors and their severity.

digit: One of the characters 0-9.

disk: A flat, circular plate with a magnetic surface on which data can be stored.

disk drive: A device that reads data from or writes data on a disk.

Disk file: A group of related records stored on disk.

diskette: A thin, flexible magnetic disk permanently enclosed in a semirigid protective jacket.

documentation: A written explanation of a program, its use, function, and operations.

edit: To punctuate a numeric field by suppressing zeros and inserting commas, decimal points, dollar signs, or other constant information.

edit code: A number or letter indicating that editing of a numeric field should be done according to a predefined pattern. This includes zero suppression and punctuation.

eighty-column card: A punch card with 80 vertical columns representing 80 characters.

end of file: The end of records in a file.

error message: See *diagnostic message*.

execute: To process input data files according to machine language instructions to produce the desired output.

factor: In RPG II programming, a field name or constant used in a calculation operation.

field: One or more adjacent record positions which contain related information.

field indicator: An indicator used to show whether a given field in an input record is plus, minus, zero, or blank.

field length: The number of positions allowed for a given field, determined by the maximum length of information that will be entered in the field.

field name: In RPG II programming, a combination of no more than six alphabetic or numeric characters (the first of which must be alphabetic) that identifies a field.

file: An organized collection of related records.

file name: In RPG II programming, a combination of no more than eight alphabetic or numeric characters (the first of which must be alphabetic) that identifies a file.

first page indicator: An indicator used to specify which lines (such as headings) should be printed on the first page only.

half adjust: A method of rounding off a number by adjusting the last digit to be kept. When the number to the right of the last digit to be retained is 5 or greater, 1 is added to the last retained digit. For example, 2.475 half adjusted to two decimal places becomes 2.48, but 2.474 becomes 2.47.

heading: A constant, usually printed at the top of a page, identifying the information or report on that page.

indicator: (1) A 2-digit or 2-character entry on the RPG II specification forms used to tell when certain calculation or output operations are to be performed. (2) An internal switch used by the object program to remember when a certain event occurs.

input: Data that is to be operated on (processed) by the system.

input file: A set of records a program uses as a source of data.

input specifications form: A coding form used to identify the different types of records in each input file and to describe the fields in each record.

instruction: A statement that specifies an operation to be performed by the system and the locations in storage of all data involved in that operation.

Keyboard: A device used to enter data into a system.

Keypunch: A device, similar to a typewriter, used for punching information into cards.

last record indicator: An indicator that signifies when the last data record has been processed.

machine language: A language that can be interpreted and used by a system.

ninety-six column card: A punch card with 96 vertical columns representing 96 characters. The columns are divided horizontally into thirds, such that the columns in the upper third are numbered 1-32, in the middle third, 33-64, and in the lower third, 65-96.

numeric: Any combination of the digits 0-9.

numeric constant: A constant used to represent a number that can consist of a decimal point, a sign, and the digits 0-9.

object program: A set of instructions in machine language. The object program is produced by the compiler from the source program.

operation: A defined action performed on one or more data items, such as adding, multiplying or comparing.

operation code: A word or abbreviation specified on the Calculation Specifications form to identify an operation such as, SUB for subtract or ADD for addition.

OR relationship: Specifying conditioning indicators so that the operation conditioned is performed when either one or both of the conditions are met.

output: Data transferred from storage to an external medium such as printed form, punched cards, or disk.

output file: A set of records that is written, punched, or printed by the system to an external medium.

Output Specifications form: An RPG II coding form used to specify the records to be written in each output file and the format of the records.

overflow: The condition that occurs when the last line to be printed on the page has been passed.

overflow indicator: An indicator that signifies when the last line on a page has been printed or passed. It can be used to specify which lines are to be printed on the next page.

overflow line: The line specified as the last line printed on a page.

overflow page: The new page after an overflow has occurred.

primary file: The main file from which a program first reads records.

printer: An output device that records information on paper in the form of printed characters.

printer spacing chart: A form used to plan the location of data in the printer output file.

processing: To perform operations on data from an input record.

processing unit: The part of a system that controls the system and its attached devices, provides storage area for the programs and data, and performs the operations specified in the program.

Program: A set of instructions that (when stored) tells the system which operations are to be done and how to do them.

program cycle: A series of operations performed by the system for each record read.

program listing: A printout which gives information about the source program, such as source statements, diagnostic messages, indicators used, storage addresses of fields, and constants used.

punched card: See *card*.

record: A group of related fields or data items treated as a unit.

record identification code: A code placed in a record when it is created to identify that record type.

record identifying indicator: An indicator that identifies the type of record being processed during the current program cycle.

record length: The total number of positions in a record.

record types: The classification of records in a file. Records are classified according to a specific field or fields within each record. Records of the same type have the same fields in the same order and identical record identification codes.

result field: The name of a field where the outcome of arithmetic calculations is kept.

resulting indicators: An indicator that can signify (1) whether the result of a calculation is plus, minus, zero, or blank, or (2) whether a field is greater than, less than, or equal to another field.

secondary file: Any file other than the primary file used in multifile processing.

Source program: A set of instructions representing a particular job as defined by the programmer. These instructions are written in a programming language, such as RPG II.

Special character: A character other than a digit, letter, or @, #, and \$. For example, *, +, and % are special characters.

Specification forms: Forms on which an RPG II program is coded and described. The four specification forms described in this manual are the Control and File Description Specifications form, the Input Specifications form, the Calculation Specifications form, and the Output Specifications form.

Storage unit: An area inside the processing unit where instructions and data are stored.

Total operations: Operations performed only after a group of records has been processed.

Total time: That part of the RPG II program cycle in which calculation and output operations specified for a group of records are done.

Zero suppression: The elimination of leading zeros in a number. For example, 00057, when zero suppressed, becomes `␣␣␣57` (␣ represents one blank space).

- address 2
- alphanumeric constants 102
- alphanumeric field 24
- AND relationship 119
- arithmetic/logic unit of system 2
- arithmetic operation 102

- blank-after 59

- calculation operations
 - description 40
 - program cycle operations 40
 - types of 42
 - sample job 49
- calculation specifications form 41
- coding 4
- comment lines 133
- compare operation 99
- compilation 5
- compiler 5
- constants
 - alphanumeric 102
 - definition 42
 - headings 66
 - numeric 42, 102
- control break 54
- control field 54
- control level indicators
 - definition 54
 - program cycle operations 56
 - RPG II specifications 57
 - sample job 60
- control section of system 2
- control specifications form 135

- data processing terms 8
- decimal positions 20, 47
- describing output records 25
- desk checking specifications 135
- detail output 15
- detail records 27
- device designation 18
- diagnostic messages 139
- documentation 133

- edit codes 34
- editing 33
- error messages 139

- factor 1 42, 99
- factor 2 42, 99
- field 8
- field indicators
 - definition 111
 - program cycle operations 111
 - RPG II specifications 112
 - sample job 114
- field location 24, 29
- field names 22, 29
- files
 - definition 8
 - input 18
 - output 18
- file description specifications form 16
- file designation 18
- file names 17, 21, 26
- first page (1P) indicator
 - definition 64
 - program cycle operations 65
 - RPG II specifications 66
 - sample job 77

- half-adjusting 47
- heading records 27

- indicators 53
- input 1
- input devices 1
- input/output operations
 - description 14
 - program cycle operations 14
 - sample job 36
- input record description 20
- input specifications form 20

job 1 36
job 2 49
job 3 60
job 4 77
job 5 89
job 6 105
job 7 114

last record (LR) indicator
 definition 74
 program cycle operations 75
 RPG II specifications 76
 sample job 77
LR indicator (see last record indicator)
L1-L9 indicators (see control level indicators)

machine language 3

numeric constants 42, 102
numeric field 24

OA-OG, OV indicators (see overflow indicators)

object program 3
OR relationship 74, 118
output 1
output devices 1
output record description 25
output specifications form 25
overflow 68
overflow handling 68
overflow indicators
 definition 68
 program cycle operations 70
 RPG II specifications 72
 sample job 77
overflow line 68
overflow page 68

primary file 18
printed output 1
printed reports 30
printer spacing chart 10
processing unit 1
 arithmetic/logic 2
 control 2
 storage 2
program 3
program cycle 11
program cycle operations 12
program listing 137
programming aids 8
programming language 3
programming terms, basic 8
programmer's job 121

record 8
record identification codes 85
record identifying indicators
 definition 81
 program cycle operations 82
 RPG II specifications 83
 sample job 89
record layout form 9
record length 19
record size 19
record type 27
record type sequence 87
reports, printed 30
result field 45
 length 47
resulting indicators
 definition 96
 program cycle operations 96
 RPG II specifications 98
 sample job 105
rounding 47
RPG II program cycle 11
RPG II programming language 3
RPG II specification forms 4

- skip 72
- skipping 31
 - tapeless carriage control 31
- source program
 - compilation 4
 - description 3
 - translation 5
- spacing
 - after printing 73
 - description 30
 - with overflow 72
- specifications forms 4
- specification form order 134
- specifying record identification codes 85
- specifying record identifying indicators 86
- specifying record type sequence 87
- storage section of system 2
- system, data processing
 - description 1
 - input devices 1
 - output devices 1
 - processing unit 1

- test the program 140
- total records 27
- type of data 24

1P indicator (see first page indicator)
01-99 indicators (see field indicators,
record identifying indicators, resulting
indicators)

READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). **No reply.**

Page Number Error

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number Comment

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Name _____

Address _____

● No postage necessary if mailed in the U.S.A.

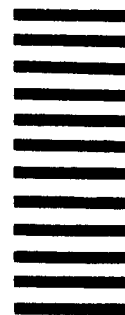
Cut Along Line

Fold

Fold

FIRST CLASS
PERMIT NO. 40
ARMONK, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES



POSTAGE WILL BE PAID BY . . .

IBM Corporation
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesota 55901

Fold

Fold



International Business Machines Corporation

**General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)**

**General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)**

Introduction to RFG report Writing (File No. S3-28) Printed in U.S.A. GC21-7514-2

READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). **No reply.**

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number *Error*

Page Number *Comment*

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Name _____

Address _____

● No postage necessary if mailed in the U.S.A.

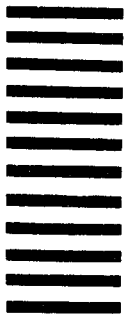
Cut Along Line

Fold

Fold

FIRST CLASS
PERMIT NO. 40
ARMONK, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES



POSTAGE WILL BE PAID BY . . .

IBM Corporation
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesota 55901

Fold

Fold



International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

Introduction to RFG
aport Writing (File No. S3-28)
Printed in U.S.A.
GC21-7514-2

READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). **No reply.**

Page Number *Error*

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number *Comment*

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Name _____

Address _____

● No postage necessary if mailed in the U.S.A.

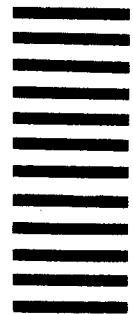
Cut Along Line

Fold

Fold

FIRST CLASS
PERMIT NO. 40
ARMONK, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES



POSTAGE WILL BE PAID BY . . .

IBM Corporation
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesota 55901

Introduction to RPG port Writing (File No. S3-28) Printed in U.S.A. GC21-7514-2

Fold

Fold



International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)



International Business Machines Corporation

**General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)**

**General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)**

Introduction to RPG II Report Writing (File No. S3-28) Printed in U.S.A. GC21-7514-2

GC21-7514-2