

**IBM System/3  
RPG II Disk File Processing  
Programmer's Guide**

GC21-7566-1  
File No. S3-28



**IBM System/3  
RPG II Disk File Processing  
Programmer's Guide**

## Second Edition (July 1974)

This is a major revision of, and obsoletes, GC21-7566-0 and Technical Newsletter GN21-7657. This manual has been extensively revised and should be reviewed in its entirety.

Information concerning IBM System/3 Model 15 has been added to the manual. The entire section entitled *Chapter 5. Using Console Devices* has been removed from this manual. The KEY (Model 6 only) and DSPLY operation codes, formerly described in Chapter 5, are described in your RPG II reference manual. For information concerning the RPG II Inquiry facility (Rollout/Rollin), also formerly described in Chapter 5, see *Related Publications*, listed in the *Preface*.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, refer to the *IBM System/3 Bibliography*, GC20-8080, for the editions that are applicable and current.

Use this publication only for the purposes stated in the *Preface*.

Publications are not stocked at the address below. Requests for copies of IBM publications and for technical information about the system should be made to your IBM representative or to the IBM 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. Comments become the property of IBM.

## Preface

This manual explains the RPG II specifications necessary to process disk files using the IBM System/3 Model 6, Model 10 Disk System, and Model 15. The following types of files are discussed:

- Sequential
- Indexed
- Direct
- Record Address

Included are examples of creating and maintaining each of these file types. For information on how this publication should be read, see *How to Use This Publication*, following the table of contents.

This manual is intended for programmers who have a basic knowledge of the RPG II language, including the ability to describe sequential and indexed files as input files. The reader must be familiar with the information presented in the *IBM System/3 Disk Concepts and Planning Guide*, GC21-7571.

*Note:* For detailed information concerning multi-volume disk files — concepts; Operation Control Language considerations; RPG II processing — see the publications listed under *Related Publications* which are appropriate for your System/3 model.

### Related Publications

These publications are recommended for additional reference:

#### *General System/3*

- *IBM System/3 Disk Concepts and Planning Guide*, GC21-7571

#### *RPG II References*

- *IBM System/3 Model 6 RPG II Reference Manual*, SC21-7517
- *IBM System/3 RPG II Reference Manual*, SC21-7504

#### *OCL References*

- *IBM System/3 Models 8 and 10 System Control Programming Reference Manual*, GC21-7512
- *IBM System/3 Model 15 System Control Programming Reference Manual*, GC21-5077
- *IBM System/3 Models 4 and 6 Operation Control Language and Disk Utility Programs Reference Manual*, GC21-7516
- *IBM System/3 Model 15 System Control Programming Concepts and Reference Manual*, GC21-5162

## Contents

<b>CHAPTER 1. SEQUENTIAL FILES . . . . .</b>	<b>1</b>
Creating a Sequential File . . . . .	1
Input Records . . . . .	3
Output Record . . . . .	3
Maintaining a Sequential File . . . . .	6
Adding Records . . . . .	7
Tagging Records for Deletion . . . . .	16
Updating Records . . . . .	16
Reorganizing a File . . . . .	22
<b>CHAPTER 2. INDEXED FILES . . . . .</b>	<b>26</b>
Creating an Indexed File . . . . .	26
Creating an Ordered Indexed File . . . . .	27
Creating an Unordered Indexed File . . . . .	28
Input Record . . . . .	29
Output Record . . . . .	29
Maintaining an Indexed File . . . . .	31
Updating Records . . . . .	32
Adding Records . . . . .	37
Tagging Records for Deletion . . . . .	49
Reorganizing a File . . . . .	49
Other Ways to Process Indexed Files . . . . .	50
Processing an Indexed File Consecutively . . . . .	50
Processing an Indexed File Randomly by Relative Record Number . . . . .	50
<b>CHAPTER 3. DIRECT FILES . . . . .</b>	<b>51</b>
Creating a Direct File . . . . .	51
Creating a Direct File without Synonyms . . . . .	53
Input Record . . . . .	54
Output Record . . . . .	54
Creating a Direct File with Synonyms . . . . .	58
Retrieval of Records in a Direct File . . . . .	60
Consecutive Retrieval . . . . .	61
Random Retrieval . . . . .	66
Maintaining a Direct File . . . . .	70
Adding Records . . . . .	70
Tagging Records for Deletion . . . . .	70
Updating Records . . . . .	71
<b>CHAPTER 4. RECORD ADDRESS FILES . . . . .</b>	<b>79</b>
Files Containing Relative Record Numbers (ADDROUT Files) . . . . .	79
Creating an ADDROUT File . . . . .	79
Processing by an ADDROUT File . . . . .	80
Files Containing Record Key Limits . . . . .	83
Creating a File with Record Key Limits . . . . .	83
Processing Sequentially within Limits . . . . .	84
<b>INDEX . . . . .</b>	<b>89</b>

## How To Use This Publication

This publication is divided into four chapters:

1. Sequential Files
2. Indexed Files
3. Direct Files
4. Record Address Files

Each of the chapters discusses the RPG II specifications needed to create and maintain a certain type of file. Since the discussions in each chapter apply to a particular file organization, you need read only the chapter for the file organization you will use.

This manual is designed to show you the entries that *must* be made in order for the system to identify the file and determine what functions are to be done. Other entries will be needed, but they depend on the job you are doing. Refer to the reference manual for your system if you need information on these additional entries.

### Note Concerning Examples

Most of the examples in this manual use MFCU input files and use 96-column card images to represent input records. This does not imply that either the MFCU or 96-column cards must be used for input; any of several input devices can be used, depending on which System/3 model and configuration you have.

If you are a Model 6 user, you would probably not use a card file for input, although you may use an online data recorder for card input. Instead, you would create sequential disk transaction files in one of the following ways:

- Using the Keyboard Data Entry conversational utility program (see *IBM System/3 Model 6 Conversational Utility Programs Reference Manual*, SC21-7528)
- Using an interactive RPG II, FORTRAN, or BASIC program to build the disk file from keyboard input





## DIRECT ACCESS STORAGE FOR MODELS 12 AND 15

The IBM 3340 Direct Access Storage Facility attaches to System/3 Model 12, to System/3 Model 15 (5704-RG1), and to System/3 Model 15 (5704-RG2).

The IBM 3344 Direct Access Storage Facility also attaches to System/3 Model 15 (5704-RG2).

Certain areas on the 3340 and 3344 disks are treated as 5444 disks. These areas, known as 5444 simulation areas, are used for the program libraries and can be also used for certain data files. The remainder of the disk space, known as main data area, can only be used for data files.

References in this manual to DISK, DISK40, and DISK45 are to be interpreted according to which disk storage device(s) is attached to the system. The following table should be used to determine the meaning of the reference:

Device Type	Model 15 (5704-RG1)	Model 12 or 15 (5704-RG1)	Model 15 (5704-RG2)
DISK	5444 Disk Storage Drive	5444 simulation area on 3340	5444 simulation area on 3340 or 3344
DISK40	Not applicable	Main data area on 3340	Main data area on 3340 or 3344
DISK45	5445 Disk Storage	Main data area on 3340	Main data area on 3340 or 3344

### IBM System/3 5448 Disk Storage Drive

The IBM System/3 5448 Disk Storage Drive on System/3 Models 8 and 10 uses the same program product support as the IBM 5445 Disk Storage. However, a separate system control program feature is required for the 5448. In general, references to 5445 in this manual also apply to 5448. For specific information about 5448 operating characteristics and programming support, see *IBM System/3 5448 Disk Storage Drive Program Reference Manual*, GC21-5168.



## Chapter 1. Sequential Files

A sequential disk file is a group of records arranged in a particular sequence and processed consecutively, one after another in the order they occur.

This chapter explains how to use the RPG II language to create and maintain a sequential file. Sample jobs are used to illustrate these functions.

To understand the sample jobs, a basic knowledge of RPG II is necessary. If you do not fully recall some of the coding used in the sample jobs, you should refer to the *IBM System/3 RPG II Reference Manual*, SC21-7504 (for Model 10 Disk System or Model 15), or the *IBM System/3 Model 6 RPG II Reference Manual*, SC21-7517, depending on the system you have.

### CREATING A SEQUENTIAL FILE

To create a sequential file, you make certain entries on the File Description sheet. The following entries are required to describe various characteristics of the disk file:

**File Description Specifications**

Column	Field	Value
1-6	File Name	
7-14	File Designation	
15	File Type	O
16-18	Sequence	F
19	Block Length	
20-23	Record Length	
24-27	Record Address Type	
28-31	Type of File	
32-35	Organization or Additional Area	
36-39	Queueing Indicator	
40-43	Key Field Starting Location	
44-47	Device	DISK
48-51	Symbl Descr	

The disk filename must be entered in columns 7 through 14. Column 15 must contain an O to indicate the file is an output file.

All records in a file must be the same length. Thus, column 19 must contain an F to specify that the record length is fixed.

A number that is equal to or a multiple of the disk record length must be entered in columns 20 through 23. This entry determines the size of the input/output area allocated by RPG II. For an explanation on block length calculation, see the *IBM System/3 Disk Concepts and Planning Guide*, GC21-7571. If you want block length calculated for you by RPG II, assign a block length equal to the record length. By blocking disk records you can increase the input/output efficiency of your program by reducing the number of accesses. You must be sure, however, that enough main storage is available for your input/output area.

Columns 24 through 27 must contain the length of the disk record. Whenever a disk file is being described, DISK (Model 6, 10, 15), DISK40 (Model 12 or 15), or DISK45 (Model 10, 12, or 15) is required in the Device columns.

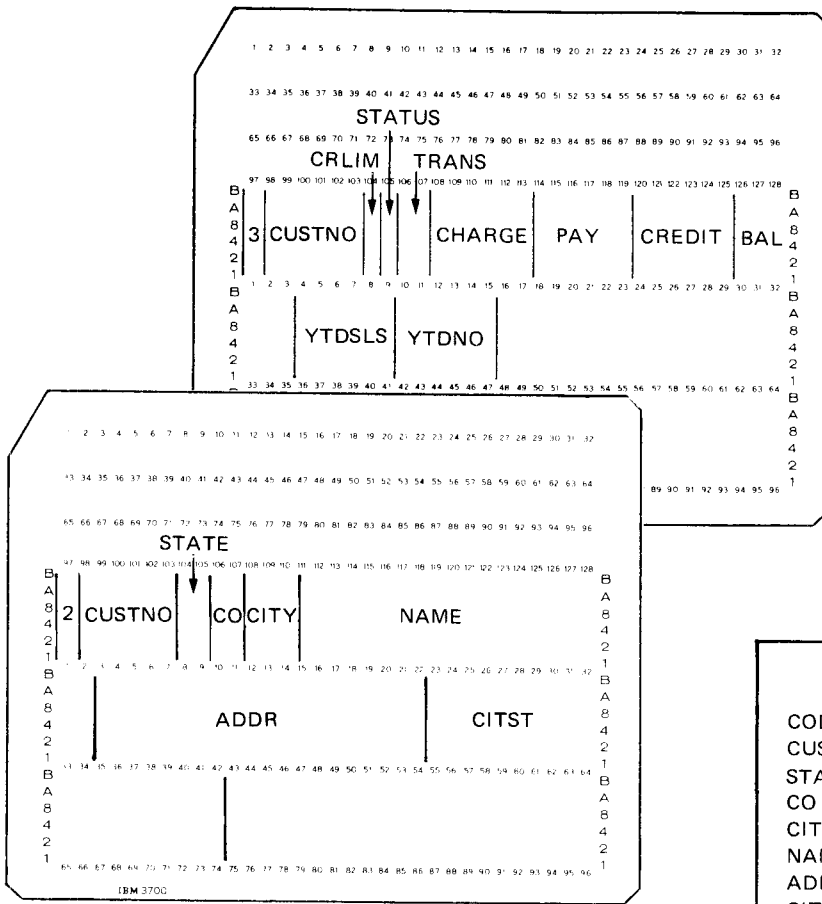
### *Example of Creating a Sequential File*

Suppose you want to create a customer file on disk. Customer numbers are to be assigned on a sequential basis; new customers are assigned the next higher number. The file will be used to produce monthly reports of each customer's status. Thus, a sequential file will serve your needs.

To create the sequential file, you must first determine the input record format and the output record format (Figure 1). The file is created by writing the customer data from the input records onto disk. Notice that space is provided on the output record so additional information can be added to the record later if necessary. Basic information about each customer in the file is also printed in the report shown in Figure 2.

Figure 3 shows the RPG II coding necessary to create the sequential customer file and print the report.

# Input Records



Key	
CODE	= Output code (CM)
CUSTNO	= Customer number
STATE	= State
CO	= County
CITY	= City
NAME	= Customer name
ADDR	= Customer street address
CITST	= City and state
CRLIM	= Credit limit
STATUS	= Status
TRANS	= Number of transactions this month
CHARGE	= Current month charges
PAY	= Current month payments
CREDIT	= Current month credits
BAL	= Balance
YTDLS	= Year-to-date sales amount
YTDNO	= Year-to-date number of sales
DELETE	= Output record code

Note: The input file need not be a card file. See *How to Use This Manual, Note Concerning Examples* at the beginning of this manual.

# Output Record

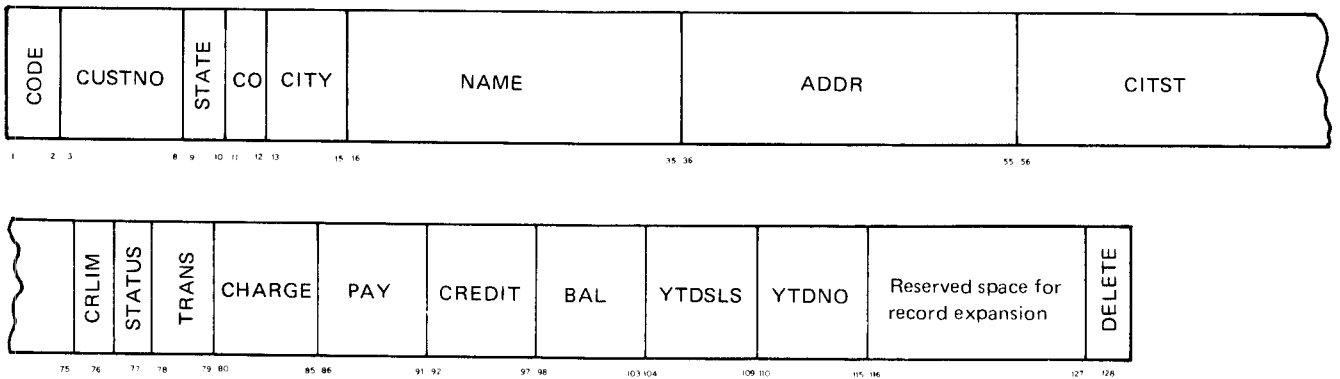


Figure 1. Input Record and Output Record Formats



RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092-1 UM-050\*  
Printed in U.S.A.

IBM International Business Machine Corporation

Program	Punching Instruction	Graphic	Card Electro Number
Programmer	Date	Punch	

Page 01 of 12 Program Identification 75 76 77 78 79 80

Control Card Specifications

Line	Form Type	Core Size to Compile	Object Output Listing Options	Core Size to Execute	Default MFCM Stacking Sequence	Inventoried Print	Number of Print Positions	Alternate Collating Sequence	Model 20	Model 20	Refer to the specific System Reference Library manual for actual entries.
01	H										

File Description Specification

Line	Form Type	Filename	File Type	File Designation	Mode of Processing	Device	Symbolic Device	Name of Label Exit	Extent Exit for DAM	File Addition/Unordered
02	F	INFILE	IP	F 96 96		MFCU1				
03	F	SEQDISK	O	F 256 128		DISK				
04	F	CUSTLISTO	F	F 74 74	OF	PRINTER				

Records are blocked (128 x 2 = 256)

RPG INPUT SPECIFICATIONS

GX21-9094-2 UM-050\*  
Printed in U.S.A.

IBM International Business Machine Corporation

Program	Punching Instruction	Graphic	Card Electro Number
Programmer	Date	Punch	

Page 02 of 12 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence	Record Identifying Indicator	Record Identification Codes	Field Location	Field Name	Field Indicators
01	I	INFILE	011 20	1 D2		2 7	CUSTNO	
02	I					15 34	NAME	
03	I					35 54	ADDR	
04	I					55 74	CITST	
05	I					8 74	DATA2	
07	I		021 30	1 D3		2 7	CUSTNO	
08	I					8 47	DATA3	

On lines 01 and 07, columns 15 - 17 contain information used to sequence check the input record. In columns 15 and 16, 01 means that record type 2 must be first followed by record type 3 (identified by 02 sequence). The 1 in column 17 means that one record type 2 and one record type 3 exists.

Figure 3 (Part 1 of 2). Creating a Sequential Customer File

**RPG OUTPUT SPECIFICATIONS**

IBM International Business Machine Corporation GX21 9090 2 U/M 050\*  
Printed in U.S.A.

Program \_\_\_\_\_ Card Electro Number \_\_\_\_\_  
 Programmer \_\_\_\_\_ Date \_\_\_\_\_ Punching Instruction \_\_\_\_\_ Graphic \_\_\_\_\_ Punch \_\_\_\_\_ Page 03 of \_\_\_\_\_ Program Identification \_\_\_\_\_

Line	Form Type	Filename	Type (H/D/T/E)		Space		Skip		Output Indicators			Field Name	End Position in Output Record	P.B.L.R.	Constant or Edit Word
			A	D	Before	After	Before	After	Not	And	And				
01	O	SEQDISK	D								30		2		
02	O												8	'CM'	
03	O												8		
04	O												15		
05	O												15		
06	O	CUSTLIST													
07	O	OR													
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														
01	O														
02	O														
03	O														
04	O														
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														

Since both input record types are needed to write a disk record, we don't want to write it until input record type 3 is processed. Indicator 30 specifies that the disk record is written after input record type 3 is processed.

CM is added to the disk record. This code is a record code that can be used to identify a customer master record in other programs.

Figure 3 (Part 2 of 2). Creating a Sequential Customer File

### MAINTAINING A SEQUENTIAL FILE

Once a file has been created, it usually needs to be maintained. File maintenance means performing those functions that keep a file current for daily processing needs. Four common file maintenance functions apply to sequential files:

1. Adding records.
2. Tagging records for deletion.
3. Updating records.
4. Reorganizing a file.









RPG CALCULATION SPECIFICATIONS

Date \_\_\_\_\_

Program \_\_\_\_\_

Programmer \_\_\_\_\_

Punching Instruction	Graphic								
	Punch								

Page 03

Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level (U, L, R, S, R)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments	
			And	And	Not								Arithmetic	Plus	Minus		Zero
01	C		30													0304	
02	C		30			CRLIM											0102
03	C		30	01	02	CRLIM											0304
04	C																

RPG OUTPUT - FORMAT SPECIFICATIONS

Date \_\_\_\_\_

Program \_\_\_\_\_

Programmer \_\_\_\_\_

Punching Instruction	Graphic								
	Punch								

Page 04

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (H/D/T/E)	Space	Skip	Output Indicators			Field Name	Edit Codes	Sterling Sign Position
						Not	And	And			
01	O	NEWLIST	H	201				1P			
02	O		OR					0V			
03	O										
04	O		H	1				1P			
05	O		OR					0V			
06	O										
07	O										
08	O										
09	O		H	2				1P			
10	O		OR					0V			
11	O										
12	O										
13	O										
14	O										
15	O										

Edit Codes							
Commas	Zero Balances to Print	No Sign	CR	-	X	Remove Plus Sign	
Yes	Yes	1	A	J	Y	Date	
Yes	No	2	B	K	Z	Field Edit	
No	Yes	3	C	L		Zero	
No	No	4	D	M		Suppress	

Constant or Edit Word

Figure 5 (Part 2 of 3). Adding Records to the End of the Sequential Customer File



### *Merging Records Between Records in the File*

Often records must be added between existing records in a sequential disk file. When records must be added in this manner, you must sort the new records and re-create the file. This new file contains the added records merged in correct order with the records from the original file.

### *Example of Merging Records Between Records in the File*

Figure 6 shows the RPG II coding necessary to merge records in the file. This example is similar to the *Example of Adding Records to the End of the File* in that the input and output records are the same format and the same report is printed.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Program \_\_\_\_\_  
 Programmer \_\_\_\_\_ Date \_\_\_\_\_  
 Puncturing Instruction \_\_\_\_\_ Graphic \_\_\_\_\_ Card Electro Number \_\_\_\_\_  
 Punch \_\_\_\_\_

Page 01 of 2 Program Identification 75 76 77 78 79 80

Control Card Specifications

Line	Form Type	Core Size to Compile	Object Output Listing Options	Core Size to Execute	Debug	MFCM Stacking Sequence	Inverted Print	Number of Print Positions	Address to Start	Model 20	Model 20	Work Tapes	Overlay Open	Overlay Printer	Binary Search	Table Error	2152 Checking	Inhibit	Read/Write/Compute	Keyboard Output	Sign Handling	IP Forms Position	Indicator Setting	File Translation	Run MFCU Zeros	Nonprint Characters	Table Load Halt	Shared I/O	Field Print	Formatted Core Dump	RPG to RPG II Conversion	Refer to the specific System Reference Library manual for actual entries.
01	H																															

File Description Specification

Line	Form Type	Filename	File Type		File Designation		Mode of Processing		Device	Symbolic Device	Name of Label Exit	Extent Exit for DAM	File Addition/Unordered	
			File Format	Block Length	Record Length	Record Address Type	Number of Tracks for Cylinder Overflow	Number of Extents						
02	F	INFILE	CS	AF	96	96			MFCU1					
03	F	DISKIN	IP	AF	256	128			DISK					
04	F	NEWDISK	O	F	256	128			DISK					
05	F	NEWLIST	O	F	82	82	OV		PRINTER					

RPG INPUT SPECIFICATIONS

Date \_\_\_\_\_  
 Program \_\_\_\_\_  
 Programmer \_\_\_\_\_

Puncturing Instruction \_\_\_\_\_ Graphic \_\_\_\_\_  
 Punch \_\_\_\_\_

Page 02 of 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence	Number (N) Option (O)	Record Identification Codes			Field Location		Field Name	Control Level (L1, L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position
					Position	Not (N) C/Z/D	Character	From	To					Plus	Minus	Zero or Blank	
01	I	INFILE	011	20	1	D2											
02	I							2	7	CUSTNO	M1						
03	I							8	9	STATE							
04	I							10	11	CO							
05	I							12	14	CITY							
06	I							15	34	NAME							
07	I							35	54	ADDR							
08	I							55	74	CITST							
09	I		021	30	1	D3											
10	I							2	7	CUSTNO	M1						
11	I							8	8	CRLIM							
12	I							9	47	REST							
13	I	DISKIN	AA	10	1	CC	2	CM									
14	I							3	8	CUSTNO	M1						
15	I							1	128	RECORD							

Figure 6 (Part 1 of 3). Merging Records Between Records in the Sequential Customer File

RPG CALCULATION SPECIFICATIONS

Date \_\_\_\_\_ Page 03 of 2 Program Identification 75 76 77 78 79 80

Punching Instruction:  Graphic  Punch

Programmer \_\_\_\_\_

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			Not	And	And								Arithmetic	Plus	Minus	
01	C		30												0304	
02	C		30			CRLIM		COMP '2'							0102	
03	C		30	01	02	CRLIM		COMP '4'							0304	
04	C															
05	C															
06	C															

RPG OUTPUT - FORMAT SPECIFICATIONS

Date \_\_\_\_\_ Page 04 of 2 Program Identification 75 76 77 78 79 80

Punching Instruction:  Graphic  Punch

Programmer \_\_\_\_\_

Line	Form Type	Filename	Type (H/D/T/E)	Space	Skip	Output Indicators			Field Name	Edit Codes	Constant or Edit Word	Sterling Sign Position
						Not	And	And				
01	O	NEWLIST	H	20	1							
02	O		OR									
03	O									47	'NEW CUSTOMER LISTING'	
04	O		H	1						8	'CUSTOMER'	
05	O		OR							24	'CUSTOMER'	
06	O									82	'CREDIT'	
07	O											
08	O											
09	O		H	2						7	'NUMBER'	
10	O		OR							22	'NAME'	
11	O									45	'ADDRESS'	
12	O									68	'CITY/STATE'	
13	O									81	'LIMIT'	
14	O											
15	O											
16	O		D	2								
17	O											
18	O											
19	O											
20	O											

Records are added to the output disk file when there is no match between the customer number in the input records and the disk file. The added records are listed.

Figure 6 (Part 2 of 3). Merging Records Between Records in the Sequential Customer File



**IBM** International Business Machines Corporation Form X21-9090 Printed in U.S.A.

**RPG OUTPUT - FORMAT SPECIFICATIONS**

Date \_\_\_\_\_ Program \_\_\_\_\_ Programmer \_\_\_\_\_

Page 05 of 2 Program Identification \_\_\_\_\_

Line	Form Type	Filename	Space			Skip			Output Indicators			Field Name	Edit Codes	End Position in Output Record	Blank After (B)	P = Packed (B = Binary)	Edit Codes							Sterling Sign Position		
			Before	After	Before	After	Not	Not	Not	Commas	Zero Balances to Print						No Sign	CR	-	X	Remove Plus Sign					
01	O																Yes	Yes	1	A	J	X	Remove Plus Sign			
02	O																Yes	No	2	B	K	Y	Date			
03	O																No	Yes	3	C	L	Z	Field Edit			
04	O																No	No	4	D	M	-	Zero Suppress			
05	O																Constant or Edit Word									
06	C																									
07	O																									
08	O	INFILE																								
09	O	NEWDISK																								
10	O																									
11	O																									
12	O																									
13	O																									
14	O																									
15	O																									
16	O																									
17	O																									
18	O																									
19	O																									
20	O																									

The input records which match the disk file on customer number are in error. They should not match because they are new records.

Input records which are in error are stacked in stacker 2. If a different input device is used, a different method of handling the error records can be used, such as printing a special message or issuing a message to the input device.

**IBM** International Business Machines Corporation Form X21-9090 Printed in U.S.A.

**RPG OUTPUT - FORMAT SPECIFICATIONS**

Date \_\_\_\_\_ Program \_\_\_\_\_ Programmer \_\_\_\_\_

Page 06 of 2 Program Identification \_\_\_\_\_

Line	Form Type	Filename	Space			Skip			Output Indicators			Field Name	Edit Codes	End Position in Output Record	Blank After (B)	P = Packed (B = Binary)	Edit Codes							Sterling Sign Position		
			Before	After	Before	After	Not	Not	Not	Commas	Zero Balances to Print						No Sign	CR	-	X	Remove Plus Sign					
01	O																Yes	Yes	1	A	J	X	Remove Plus Sign			
02	C																Yes	No	2	B	K	Y	Date			
03	O																No	Yes	3	C	L	Z	Field Edit			
04	O																No	No	4	D	M	-	Zero Suppress			
05	O																Constant or Edit Word									
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																									

Copy the record from the original file.

Figure 6 (Part 3 of 3). Merging Records Between Records in the Sequential Customer File



Since updating means getting a record from a disk file, changing some data, and putting the record back in its original location, an update file is like a combination input/output file. For this reason, the file to be updated must be specified on both the Input and Output-Format sheets. Field locations should agree between the two sheets. Field names may vary depending on the updating being done. Field lengths must agree.

International Business Machines Corporation

### RPG INPUT SPECIFICATIONS

Punching Instruction	Graphic								
	Punch								

Page 1 2

Identification Codes												Field Location		Field Name	Column Level (L-L9)																		
2						3						From	To																				
Position	Not (N) C/Z/O	Character	Position	Not (N) C/Z/D	Character	Blank or Street P = Package/B = Binary																											
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		
																	40	46	46														

International Business Machines Corporation

### RPG OUTPUT - FORMAT SPECIFICATIONS

Punching Instruction	Graphic								
	Punch								

Page \_\_\_\_\_

Identification Codes												Field Location		Field Name	Column Level (L-L9)																		
2						3						From	To																				
Position	Not (N) C/Z/O	Character	Position	Not (N) C/Z/D	Character	Blank or Street P = Package/B = Binary																											
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		

Commas		Zero Balances to Print		Edit	
Yes	Yes	Yes	Yes	1	
Yes	No	No	Yes	2	
No	Yes	Yes	No	3	
No	No	No	No	4	

Constant 0

Should Agree

May Vary

*Example of Updating Records*

Periodically, you might want to update the accounting information for each customer in your customer file. You might also want to tag some customer records for deletion. A printed report, like the one shown in Figure 7, lists the updated information and the records tagged for deletion.

The TRANS file contains two input record types. One type identifies disk records to be deleted (D in column 1); the other type contains information needed to update the MASTER file (3 in column 1). Figure 8 shows the RPG II coding necessary to update records and to tag records for deletion.

	1	2	3	4	5	6	7	8	9	10
	1	2	3	4	5	6	7	8	9	10
H	1	2	3	4	5	6	7	8	9	10
H	3									
H	4									
D	6									
	7									
	8									
	9									
	10									

CUSTOMER FILE LISTING							
CUSTOMER NUMBER	CUSTOMER NAME	CHARGES	CREDITS	PAYMENTS	NEW BALANCE	CREDIT LIMIT	OVER LIMIT
136728	JONES VARIETY	5,000.00	3,000.00	1,500.00	500.00	1,000	
301628	JIM'S 5 AND 10	600.00	200.00	100.00	300.00	250	****
795246	SCHMIDT HARDWARE	2,500.00	1,000.00	500.00	1,000.00	2,000	

Figure 7. Report of Updated Customer Records and Deleted Customer Records



**IBM** International Business Machines Corporation Form X21-9093 Printed in U.S.A.

**RPG CALCULATION SPECIFICATIONS**

Date: \_\_\_\_\_ Page: 03 Program Identification: 75 76 77 78 79 80

Punching Instruction: \_\_\_\_\_ Graphic: \_\_\_\_\_ Punch: \_\_\_\_\_

Programmer: \_\_\_\_\_

Indicator 01 is on for an input record which deletes a disk record. Indicator 21 is set for use during update of the MASTER file.

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Compare			Comments
			Not	And	And								High	Low	Equal	
01	C		01	MR		SETON						21				SAVE INDICATOR
02	C		02	MR		TOTCHG	ADD	CHG	TOTCHG	62						ADD CHARGES
03	C		02	MR		TOTPAY	ADD	PAYMT	TOTPAY	62						ADD PAYMENTS
04	C		02	MR		TOTCRD	ADD	CRDT	TOTCRD	62						
05	C		03	MR		CHARGE	ADD	TOTCHG	CHARGE							
06	C		03	MR		PAY	ADD	TOTPAY	PAY							
07	C		03	MR		CREDIT	ADD	TOTCRD	CREDIT							
08	C		03	MR		BAL	ADD	TOTCHG	BAL							
09	C		03	MR		BAL	SUB	TOTPAY	BAL							
10	C		03	MR		BAL	SUB	TOTCRD	BAL							
11	C		03			CRLIM	COMP	'2'					1112			
12	C		03	11	12	CRLIM	COMP	'4'					1314			
13	C		03	11		BAL	COMP	250.00					20			
14	C		03	12		BAL	COMP	500.00					20			
15	C		03	13		BAL	COMP	1000.00					20			
16	C		03	14		BAL	COMP	2000.00					20			
17	C	L1				SETOF							111213			
18	C	L1				SETOF							142021			

The values in the MASTER file are updated by totals accumulated from transaction records.

Check for balance exceeding credit limit.

Set off indicators for next cycle.

**IBM** International Business Machines Corporation Form X21-9093 Printed in U.S.A.

**RPG CALCULATION SPECIFICATIONS**

Date: \_\_\_\_\_ Page: 04 Program Identification: 75 76 77 78 79 80

Punching Instruction: \_\_\_\_\_ Graphic: \_\_\_\_\_ Punch: \_\_\_\_\_

Programmer: \_\_\_\_\_

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			Not	And	And								Arithmetic	Plus	Minus	
01	C		L1			TOTCHG	SUB	TOTCHG	TOTCHG							
02	C		L1			TOTPAY	SUB	TOTPAY	TOTPAY							
03	C		L1			TOTCRD	SUB	TOTCRD	TOTCRD							
04	C															
05	C															
06	C															
07	C															
08	C															
09	C															

Clear accumulator fields for next cycle.

Figure 8 (Part 2 of 4). Updating and Deleting Records in the Sequential Customer File













## Chapter 2. Indexed Files

An indexed disk file has two parts: an index and the records. The index contains the key field and disk address of each record in the file; it is organized sequentially by key field. Records in the file are organized in the order in which they are loaded. Records in an indexed file can be processed in the following ways:

1. Sequentially by key.
2. Sequentially within limits.
3. Randomly by key or relative record number.
4. Consecutively (without keys).
5. By ADDROUT file.

This chapter explains how to use the RPG II language to create and maintain a single volume indexed file. Sample jobs are used to illustrate these functions.

To understand the sample jobs, a basic knowledge of RPG II is necessary. If you do not fully recall some of the coding used in the sample jobs, you should refer to the *IBM System/3 RPG II Reference Manual*, SC21-7504 (for the Model 10 Disk System or the Model 15), or the *IBM System/3 Model 6 RPG II Reference Manual*, SC21-7517, depending on the system you have.

### CREATING AN INDEXED FILE

You can create a single volume indexed file with records that are in ordered or unordered sequence. An ordered sequence means the records are arranged in order according to the major control field that will be used as the key of the file. If an ordered sequence is specified, records are sequence-checked by System/3 data management.

An unordered sequence means the records are in no particular order. For example, if records in an inventory item file were organized by frequency of use, they would be in an unordered sequence with the most active items at the beginning of the file. System/3 data management will not sequence check the records or check for duplicate records while an indexed file is being loaded in unordered sequence. The index of an unordered file is sorted into ascending sequence after all the records have been loaded. At this time, System/3 data management will detect duplicate records.

*Note:* Multivolume indexed files must be loaded in key field sequence, that is, in ordered sequence. See *Related Publications* in the *Preface* for sources of additional information about multivolume files.

## Creating an Ordered Indexed File

To create an indexed file in an ordered sequence, you must make the following entries on the File Description sheet:

**File Description Specifications**

File Type														Mode of Processing														Device		Symbol Device													
File Description														Length of Key Field or of Record Address Field																													
End of File														Record Address Type																													
Sequence														Type of File Organization or Additional Area																													
File Format														Key Field Starting Location																													
Block Length														Key Field Length																													
Record Length																																											
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
							O	F							A							I							DISK														

The disk filename must be entered in columns 7 through 14. Column 15 must contain an O to indicate that the file is an output file.

All records in a file must be the same length, so column 19 must contain an F to specify that the record length is fixed.

A number equal to, or a multiple of, the disk record length must be entered in columns 20 through 23. This entry determines the size of the input/output area allocated by RPG II. If you want block length equal to the record length. By blocking disk records, you can increase the input/output efficiency of your program by reducing the number of accesses. You must be sure, however, that enough main storage is available for your input/output area.

*Note:* Block length calculation and input/output area allocation are described in *IBM System/3 Disk Concepts and Planning Guide*, GC21-7571. On Model 15, RPG II uses double buffering for indexed file output; therefore, main storage must be available for at least twice the block length plus twice the index buffer length.

Columns 24 through 27 must contain the length of the disk record. Whenever a disk file is being described, DISK (Model 6, 10, 15), DISK40 (Model 12, 15), or DISK45 (Model 10, 12, 15) is required in the Device columns.

Indexed files are allowed only on the main data area, and are not allowed on the simulation area. Models with 5445 or 3340 drives must use DISK45 and DISK40 respectively.

Entries in columns 31 and 32 indicate that an indexed file is to be created. An I in column 32 specifies an indexed file; an A in column 31 specifies that a key field exists.

The other two entries describe the length and location of the key. Key length, entered in columns 29 through 30, is the same for all records in a file. The maximum key length is 29 characters. The location of the first character of the key is specified in columns 35 through 38. Thus, if a 6-character key field were located in positions 73 through 78 of a disk record, you would enter a 73 in columns 37 through 38 of the File Description sheet and a 6 in column 30.



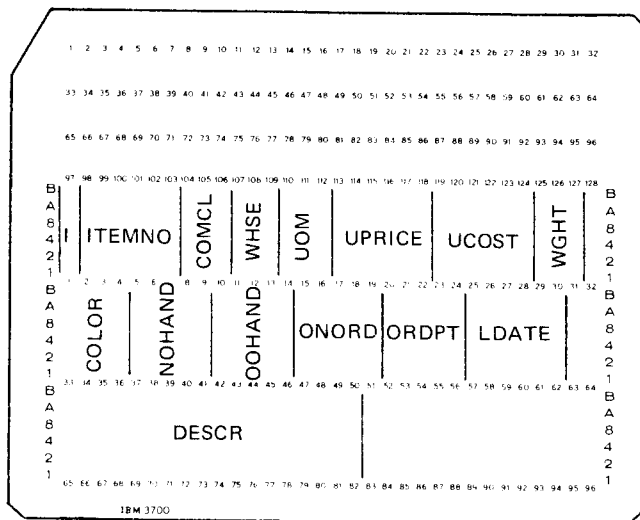
### Example of Creating an Indexed File

Suppose you want to create an inventory file on disk and want to process the records in various ways (sequentially, randomly, within limits). An indexed file provides this processing flexibility. Whether you load your records in an ordered or an unordered sequence depends on which sequence gives you the most processing efficiency.

To create an indexed file, you must first determine the input record format and the output record format (Figure 11). The file is created by writing the inventory item data from the input records onto disk. Notice that the output record format provides space so additional information can be added to the record later if necessary.

Figure 12 shows the RPG II coding necessary to create the indexed inventory file. The program will count the number of records loaded and print the total after the file has been created.

#### Input Record



Note: The input file need not be a card file. See *How to Use This Manual, Note Concerning Examples* at the beginning of this manual.

Key	
CODE	= Record identifying code (I)
ITEMNO	= Item number
COMCL	= Commodity class
WHSE	= Warehouse location
UOM	= Unit of measure
UPRICE	= Unit price
UCOST	= Unit cost
WGHT	= Weight
COLOR	= Color
NOHAND	= New on hand quantity
OOHAND	= Old on hand quantity
ONORD	= Quantity on order
ORDPT	= Order point
LDATE	= Latest transaction date
DESCR	= Description
DELETE	= Output record delete code

#### Output Record

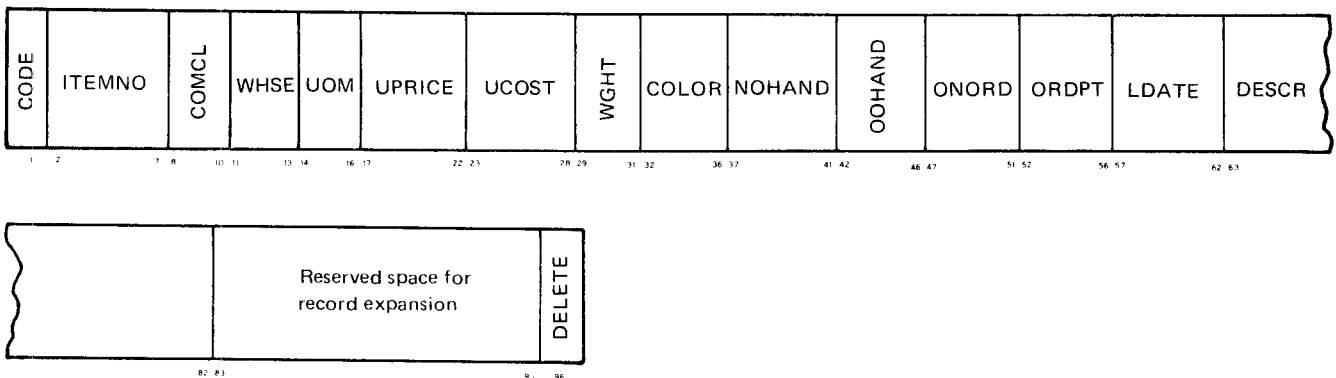


Figure 11. Input Record and Output Record Formats

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

GX21 9092 3 UM-0507  
 Printed in U.S.A.

IBM International Business Machine Corporation

Program	Punching Instruction	Graphic	Card Electro Number
Programmer	Date	Punch	

Page 01 of 2 Program Identification 75 76 77 78 79 80

Control Card Specifications

Line	Form Type	Core Size to Compile	Object Output Listing Options	Core Size to Execute	Debug M/C M Stacking Sequence	Inverted Print	360/20 2501 Buffer	Number of Print Positions	Alternate Collating Sequence	Model 20 Address to Start	Model 20 Work Tables	Model 20 Read/Write/Compare	Model 20 Keyboard Output	Model 20 Sign Handling	Model 20 IP Forms Position	Model 20 Indication Setting	Model 20 File Translation	Model 20 Punch MFCU Zeros	Model 20 Nonprint Characters	Model 20 Table Load Halt	Model 20 Shared I/O	Model 20 Field Print	Model 20 Formatted Core Dump	Model 20 RPG to RPG II Conversion	
01	H																								

Refer to the specific System Reference Library manual for actual entries.

File Description Specification

Line	Form Type	Filename	File Type	File Designation	End of File	Sequence	File Format	Block Length	Record Length	Mode of Processing	Record Address Type	Type of File Organization or Additional Area	Device	Symbolic Device	Labels S/N/E/M	Name of Label Exit	Extent Exit for DAM	Core Index	File Addition/Unordered	Number of Tracks for Cylinder Overflow	Number of Extents	Tape Rewind	File Condition U1 U8	
02	F	INFILE	I	PE	F			96	96				MFCU1											
03	F	OUTDISK	O	F	F			192	96	RO	6AI		2 DISK											U
04	F	REPORT	O	F	F			20	20				PRINTER											

Make this specification only if you want the records to be loaded in an unordered sequence.

RPG INPUT SPECIFICATIONS

GX21 9994 2 U M 0507  
 Printed in U.S.A.

IBM International Business Machine Corporation

Program	Punching Instruction	Graphic	Card Electro Number
Programmer	Date	Punch	

Page 02 of 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence	Record Identification Codes									Field Location		Field Name	Control Level (L/LB)	Matched Fields or Changing Fields	Field Record Relation	Field Indicators						
				1	2	3	Position	Character	Position	Character	Position	Character	From	To					Plus	Minus	Zero or Blank				
01	I	INFILE	AA	01																					
02	I																								
03	I																								
04	I																								
05	I																								
06	I																								

Figure 12 (Part 1 of 2). Creating an Indexed Inventory File



**IBM** International Business Machines Corporation Form X21-9093 Printed in U.S.A.

### RPG CALCULATION SPECIFICATIONS

Date \_\_\_\_\_ Page **03** Program Identification 75 76 77 78 79 80

Punching Instruction: 

Graphic					
Punch					

Programmer \_\_\_\_\_

Line	Form Type	Control Level (LO, LR, LP, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments
			Not	And	And							Arithmetic	Plus	Minus	
01	C		01			COUNT	ADD	1	COUNT	50					
02	C														

**IBM** International Business Machines Corporation Form X21-9090 Printed in U.S.A.

### RPG OUTPUT - FORMAT SPECIFICATIONS

Date \_\_\_\_\_ Page **04** Program Identification 75 76 77 78 79 80

Punching Instruction: 

Graphic					
Punch					

Programmer \_\_\_\_\_

Line	Form Type	Filename	Type (M/C/T/E)	Space Before	Skip	Output Indicators			Field Name	Edit Codes	Constant or Edit Word	Sterling Sign Position
						Space After	Not	And				
01	O	OUTDISK D				01			RECORD	96		
03	O	REPORT T		301		LR			COUNT Z	5		
05	O									20	'RECORDS LOADED'	

Edit Codes					
Commas	Zero Balances to Print	No Sign	CR	-	X = Remove Plus Sign
Yes	Yes	1	A	J	Y = Date
Yes	No	2	B	K	Z = Field Edit
No	Yes	3	C	L	- = Zero Suppress
No	No	4	D	M	

Figure 12 (Part 2 of 2). Creating an Indexed Inventory File

### MAINTAINING AN INDEXED FILE

Once a file is created, it usually needs to be maintained. File maintenance means performing those functions that keep a file current for daily processing needs. Four file maintenance functions apply to indexed files:

1. Updating records.
2. Adding records.
3. Tagging records for deletion.
4. Reorganizing a file.







*Example of Updating Records Randomly by Key*

Suppose that whenever a transaction is made you want to update the item record in the inventory file. Since the transaction records are not in sequence, you will process the file randomly by key. You might also want to tag some records for deletion while the file is being processed.

Figure 13 shows the RPG II coding necessary to update the records and tag records for deletion. The TRANS file contains the update and deletion information.

Refer to Appendix H for more information on blocking records when processing an indexed file randomly.

This page intentionally left blank







### *Updating Records Sequentially within Limits*

See Chapter 4 in this manual for information on updating records sequentially within limits.

### **Adding Records**

After a file is created, it is often necessary to add records to the file. For an indexed file, records can be added that contain keys which are: above the highest key presently in the file; below the lowest key presently in the file; or in between keys presently in the file. In any case, the added records are placed at the end of the records presently in the file. The index entry for each added record is written at the end of the current entries in the index area. After all records are added, the index is automatically sorted by the system.

If many records are to be added to the file, the time required for the index sort can be decreased by allocating a special work file. This requires no special RPG II coding but does require a special OCL statement. See the section concerning use of OCL in the manuals listed in the *Preface* under *OCL References* for additional information and an example of this option. For a further discussion of indexed file performance considerations, see *IBM System/3 Disk Concepts and Planning Guide*, GC21-7571.

You can add records to a file in one of two ways:

1. Randomly by key.
2. Sequentially by key.

### *Adding Records Randomly by Key Using Chaining*

Records can be added randomly by key to an indexed file in two ways, either with or without chaining.

*Chaining* means matching the record key of the record to be added with the keys of the index. This matching is done as a check to ensure that the record to be added is not a duplicate of a record already in the file. The data used as a record key in the chain operation can be either a field in an input record or it can be created in your program.



The entry on the Output-Format sheet is an ADD in columns 16 through 18:

International Business Machines Corporation

**RPG OUTPUT - FORMAT SPECIFIC**

Punching Instruction	Graphic
	Punch

Space	Key	Output Indicators												Field Name	End Position in Output Member																
		1	2	3	4	5	6	7	8	9	0	1	2			3	4	5	6	7	8	9	0								
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
ADD																															

This entry tells the system that the fields defined on the following specification lines constitute the record to be added to the file specified in columns 7 through 14.

*Example of Adding Records Randomly by Key Using Chaining*

Suppose you want to add new inventory items to the indexed inventory file created in the *Example of Creating an Indexed File*. The new records are not in sequence. New record keys may be lower, between, or higher than keys presently in the file.

Input and output records will be in the same format as the records used to create the file. A printed report (Figure 14) will list all the new records added to the file.

H	1	XX/XX/XX	NEW INVENTORY ITEMS	PAGE	XXXX
H	3	ITEM	WAREHOUSE	UNIT	ON
H	4	NUMBER	LOCATION	PRICE	COST
	5				HAND
D	6	XXXXXXXXXXXXXXXXXXXX	XXX	X,XXX.XX	X,XXX.XX
	7	(ITEMNO)	(WHSE)	(UPRICE)	(UCOST)
	8				(NOHAND) (ONORD)
	9				

06/18/70	NEW INVENTORY ITEMS	PAGE	1
ITEM NUMBER	DESCRIPTION	WAREHOUSE LOCATION	UNIT PRICE UNIT COST ON HAND ON ORDER
413010	CH001 BOX 100A FLUSH	768	4.90 4.00 10 6
412146	CH143 BREAKER 15A	913	.89 .59 100 75
411126	1500 TWIN SOCKET B	493	1.12 .97 500 325

**Figure 14. Report of New Items Added to the Inventory File**



**IBM** International Business Machines Corporation Form X21-9083 Printed in U.S.A.

### RPG CALCULATION SPECIFICATIONS

Date: \_\_\_\_\_ Page **03** Program Identification: \_\_\_\_\_

Programmer: \_\_\_\_\_

Line	Form Type	Control Level (L.O.L.R.S.R.)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators			Comments
			And	And	Not							Arithmetic	Compare	Lookup	
01	C		01			ITEMNO	CHAINMASTER								
02	C		01 99			COUNT	ADD 1	COUNT	50						
03	C		01N99			ERRTOT	ADD 1	ERRTOT	50						
04	C														
05	C														
06	C														
07	C														
08	C														

If 99 is on, the item is not in the file. Add the record. If 99 is off, the item has a duplicate in the file. This is considered an error in this example.

**IBM** International Business Machines Corporation Form X21-9090 Printed in U.S.A.

### RPG OUTPUT - FORMAT SPECIFICATIONS

Date: \_\_\_\_\_ Page **04** Program Identification: \_\_\_\_\_

Programmer: \_\_\_\_\_

Line	Form Type	Filename	Space			Skip			Output Indicators			Field Name	Edit Codes	Sterling Sign Position
			Before	After	Before	After	Not	Not	Not					
01	O	ADDLIST	H		201			1P						
02	O		OR					OF						
03	O										UPDATE	Y		
04	O												8	
05	O												44	'NEW INVENTORY ITEMS'
06	O												74	
07	O												69	'PAGE'
08	O		H		1			1P						
09	O		OR					OF						
10	O												5	'ITEM'
11	O												38	'WAREHOUSE'
12	O												46	'UNIT'
13	O												56	'UNIT'
14	O												64	'ON'
15	O												73	'ON'
16	O		H		2			1P						
17	O		OR					OF						
18	O												6	'NUMBER'
19	O												23	'DESCRIPTION'
20	O												37	'LOCATION'
21	O												46	'PRICE'

Figure 15 (Part 2 of 3). Adding Records to the Indexed Inventory File











RPG INPUT SPECIFICATIONS

Program \_\_\_\_\_ Punching Instruction \_\_\_\_\_ Graphic \_\_\_\_\_ Card Electro Number \_\_\_\_\_  
 Programmer \_\_\_\_\_ Date \_\_\_\_\_ Punch \_\_\_\_\_ Page 02 of 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence	Record Identification Codes			Field Location		Field Name	Control Level (L, U)	Field Indicators		
				1	2	3	From	To			Plus	Minus	Zero or Blank
0.1	I	TRANS	AA	01	1	CI							
0.2	I						2	7	ITEMNO	M1			
0.4	I						11	13	WHSE				
0.5	I						17	22	UPRICE				
0.6	I						23	28	UCST				
0.7	I						37	41	NOHAND				
0.8	I						47	51	NOORD				
0.9	I						63	82	DESCR				
1.0	I		AB	02	1	CR	1	96	RECORD				
1.1	I												
1.2	I						2	7	ITEMIN	M1			
1.3	I						8	12	QTY				
1.4	I						13	18	COST				
1.5	I		AC	03	1	INCI		24	DATE				
1.6	I												
1.7	I	MASTER	BA	04	1	CI							
1.8	I						2	7	ITEM	M1			
1.9	I						23	28	UCOST				
2.0	I						37	41	NOHAND				
2.1	I						42	46	NOHAND				
2.2	I						57	62	LDATE				
2.3	I						1	96	MASREC				

New item records

Receipts of existing inventory items

Catch-all for unidentified records

Inventory master file

RPG CALCULATION SPECIFICATIONS

Program \_\_\_\_\_ Punching Instruction \_\_\_\_\_ Graphic \_\_\_\_\_ Card Electro Number \_\_\_\_\_  
 Programmer \_\_\_\_\_ Date \_\_\_\_\_ Punch \_\_\_\_\_ Page 03 of 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level (L, U), L.F., S.R., AN/(OR)	Indicators			Factor 1	Operation	Factor 2	Result Field		Resulting Indicators	Comments
			And	And	And				Name	Length		
0.2	C		02	MR		Z-ADDCOST	UCOST					
0.3	C		02	MR		Z-ADDNOHAND	NOHAND					Update inventory master record
0.4	C		02	MR	QTY	ADD NOHAND	NOHAND					
0.5	C		02	MR		MOVE DATE	LDATE					

Figure 16 (Part 2 of 4) Adding Records Sequentially by Key





## Tagging Records for Deletion

When a record becomes inactive, you probably will not want to process it with the other records. Since the record might not be physically removed from the file, however, you must identify it so it can be bypassed. One way to identify the record is to put a code, called a delete code, in a particular location in the record. This code can be any character you want. Any program which should not process deleted records can check for this code. If the code is present, the program bypasses the record. The deleted records can be physically removed from the file by reorganizing the file (see *Reorganizing a File* in this section).

### *Example of Tagging Records for Deletion*

An example of tagging records for deletion is shown in this section under *Updating Records, Example of Updating Records Randomly by Key*.

## Reorganizing a File

Reorganizing a file is similar to creating the file. Reorganization may be necessary for two reasons:

1. To increase processing efficiency.
2. To free disk space.

You can increase processing efficiency by restoring your file to its original sequence. When you add records to a file, these records are added at the end of the records already in the file; however, the keys are always in order in the index. When the file is processed sequentially by key, the disk access arm moves back and forth between the sequenced records (those originally created) and the added records. This increases processing time.

Disk space can be made available by removing inactive records during a file copy. All records with a delete code can be physically deleted from the file.

If you want your file in an ordered sequence, you can reorganize the file to place the added records in sequence with the records originally created and remove records tagged for deletion. To reorganize the file in this manner, you can use the IBM Disk Copy/Dump program. For an explanation on how to use this program, see the *IBM System/3 Model 10 Disk System Control Programming Reference Manual*, GC21-7512, the *IBM System/3 Model 15 System Control Programming Reference Manual*, GC21-5077, or the *IBM System/3 Model 6 Operation Control Language and Disk Utility Programs Reference Manual*, GC21-7516, depending on the system you have.

If you want your file in an unordered sequence and want only to delete records in the file, you can also use the IBM Disk Copy/Dump program.

When you want to change the order of the records and maintain an unordered sequence, you must determine what method can be used to produce the file in the required order. For example, if a count of activity was maintained in each master record, the file could be sorted in descending sequence on this activity field, then reloaded as an indexed file. This would place the most active records at the front of the file.

## **OTHER WAYS TO PROCESS INDEXED FILES**

### **Processing an Indexed File Consecutively**

An indexed file may be processed consecutively (read only) by defining the indexed file as a sequential input file in the File Description Specifications. When an indexed file is processed consecutively, the file index is bypassed and data records are read consecutively from the beginning of the file to the end, exactly as a sequential file. Indexed files may not be created, added to, or updated consecutively.

Consecutive processing of an indexed file is useful, for example, for reading records from an indexed file when the file index is unusable for some reason.

### **Processing an Indexed File Randomly by Relative Record Number**

An indexed file may be processed randomly by relative record number if the file is an input file. The file must be described as a sequential file (that is, columns 31 and 32 must be blank) and must be described as a chained file (C in column 16 of the File Description sheet). The CHAIN operation code must be used in calculations to read records from the file. Records may not be updated, added, or written out to an indexed file using this method.

Random processing of an indexed file by relative record number can provide improved performance over random processing by key, because the file index need not be read. However, it is the user's responsibility to ensure that records in the indexed file are properly sequenced for this type of processing.



Columns 24 through 27 must contain the length of the disk record. Column 28 contains an R to indicate that random processing is to take place. Whenever a disk file is being described, DISK (Model 6, 10, 15), DISK40 (Model 12 or 15), or DISK45 (Model 10, 12, or 15) is required in columns 40 through 46.

Relative record numbers are always used with the CHAIN operation code in your program to make the corresponding record locations in a direct file available for loading. The data used as a relative record number in the chain operation can be either a field in an input record or it can be created in your program. To use the chain operation, you must make the following entries on the Calculation sheet:

International Business Machines Corporation

### RPG CALCULATION SPECIFICATIONS

Punching Instruction	Graphic Punch						
----------------------	---------------	--	--	--	--	--	--

Page 

1	2
---	---

\$	Factor 1	Operation	Factor 2	Result Field	Field Length	Resulting Indicators
16						Arithmetic
17						Plus Minus Zero
18						Compare
19						High Low Equal
20						1 > 2 1 < 2 1 = 2
21						Lookup
22						Table Factor 01-1
23						High Low Equal
24						54 55
25						56 57
26						58 59
27						60 61
28						62 63
29						64 65
30						66 67
31						68 69
32						70 71
33						72 73
34						74 75
35						76 77
36						78 79
37						80 81
38						82 83
39						84 85
40						86 87
41						88 89
42						90 91
43						92 93
44						94 95
45						96 97
46						98 99
47						100 101
48						102 103
49						104 105
50						106 107
51						108 109
52						110 111
53						112 113
54						114 115
55						116 117
56						118 119
57						120 121
58						122 123
59						124 125
60						126 127
61						128 129
62						130 131
63						132 133
64						134 135
65						136 137
66						138 139
67						140 141
68						142 143
69						144 145
70						146 147
71						148 149
72						150 151
73						152 153
74						154 155
75						156 157
76						158 159
77						160 161
78						162 163
79						164 165
80						166 167
81						168 169
82						170 171
83						172 173
84						174 175
85						176 177
86						178 179
87						180 181
88						182 183
89						184 185
90						186 187
91						188 189
92						190 191
93						192 193
94						194 195
95						196 197
96						198 199
97						200 201
98						202 203
99						204 205
100						206 207
101						208 209
102						210 211
103						212 213
104						214 215
105						216 217
106						218 219
107						220 221
108						222 223
109						224 225
110						226 227
111						228 229
112						230 231
113						232 233
114						234 235
115						236 237
116						238 239
117						240 241
118						242 243
119						244 245
120						246 247
121						248 249
122						250 251
123						252 253
124						254 255
125						256 257
126						258 259
127						260 261
128						262 263
129						264 265
130						266 267
131						268 269
132						270 271
133						272 273
134						274 275
135						276 277
136						278 279
137						280 281
138						282 283
139						284 285
140						286 287
141						288 289
142						290 291
143						292 293
144						294 295
145						296 297
146						298 299
147						300 301
148						302 303
149						304 305
150						306 307
151						308 309
152						310 311
153						312 313
154						314 315
155						316 317
156						318 319
157						320 321
158						322 323
159						324 325
160						326 327
161						328 329
162						330 331
163						332 333
164						334 335
165						336 337
166						338 339
167						340 341
168						342 343
169						344 345
170						346 347
171						348 349
172						350 351
173						352 353
174						354 355
175						356 357
176						358 359
177						360 361
178						362 363
179						364 365
180						366 367
181						368 369
182						370 371
183						372 373
184						374 375
185						376 377
186						378 379
187						380 381
188						382 383
189						384 385
190						386 387
191						388 389
192						390 391
193						392 393
194						394 395
195						396 397
196						398 399
197						400 401
198						402 403
199						404 405
200						406 407
201						408 409
202						410 411
203						412 413
204						414 415
205						416 417
206						418 419
207						420 421
208						422 423
209						424 425
210						426 427
211						428 429
212						430 431
213						432 433
214						434 435
215						436 437
216						438 439
217						440 441
218						442 443
219						444 445
220						446 447
221						448 449
222						450 451
223						452 453
224						454 455
225						456 457
226						458 459
227						460 461
228						462 463
229						464 465
230						466 467
231						468 469
232						470 471
233						472 473
234						474 4



## Creating a Direct File without Synonyms

A direct file can be created without synonyms when the relative record number either corresponds to a field containing sequential values or is derived in such a way that no synonyms are produced.

When you do not have synonyms, you can load records into a direct file in a single pass. You do this by specifying a chained output file and writing records in the file by means of the chain operation. Record locations cannot be inspected before they are filled with data. If a synonym is encountered, it is written over the previous record. The previous record is lost.

### *Example of Creating a Direct File without Synonyms*

Suppose you want to create a customer file on disk. The following are significant characteristics of the file:

- Customer numbers are assigned on a sequential basis; new customers are assigned the next higher number.
- Deletions from the file are few.
- The file will be used to process invoices, orders, and cash payments in a random manner.
- The file must allow direct inquiry to any customer's record.
- The file has low activity; for example, out of 5000 customer records, only 100 invoices are processed per day.

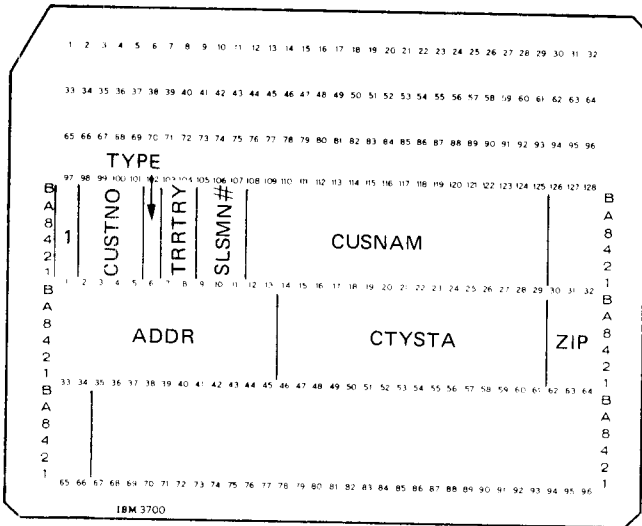
You need both the direct and consecutive processing capabilities offered by indexed and direct file organizations. Because the customer numbers are assigned consecutively, synonym records are not a consideration. For this reason, and because there will be few deletions from the file creating wasted space, direct file organization provides maximum flexibility and access speed.

Your first step, then, is to create the direct file. The record format shown in Figure 16 satisfies your information needs. Additional fields in the record will contain information to be used in specific jobs, such as customer payments, invoicing, and sales analysis. (Various applications using the customer file are described later in this chapter.)

The file is created from data on input records (Figure 17). The customer number (CUSTNO) is used as the relative record number to chain to the direct file. The customer data from the input records is then written on disk. Each record written on disk is also printed in the report shown in Figure 18.

Figure 19 shows the RPG II coding necessary to create the direct customer file.

### Input Record



Key	
IDCODE	= Identification code
CUSTNO	= Customer number
TYPE	= Type
TRRTRY	= Territory
SLSMN#	= Salesman number
CUSNAM	= Customer name
ADDR	= Customer street address
CTYSTA	= City and state
ZIP	= Zip code
DELETE	= Output record delete code

*Note:* The input file need not be a card file. See *How to Use This Manual, Note Concerning Examples* at the beginning of this manual.

### Output Record

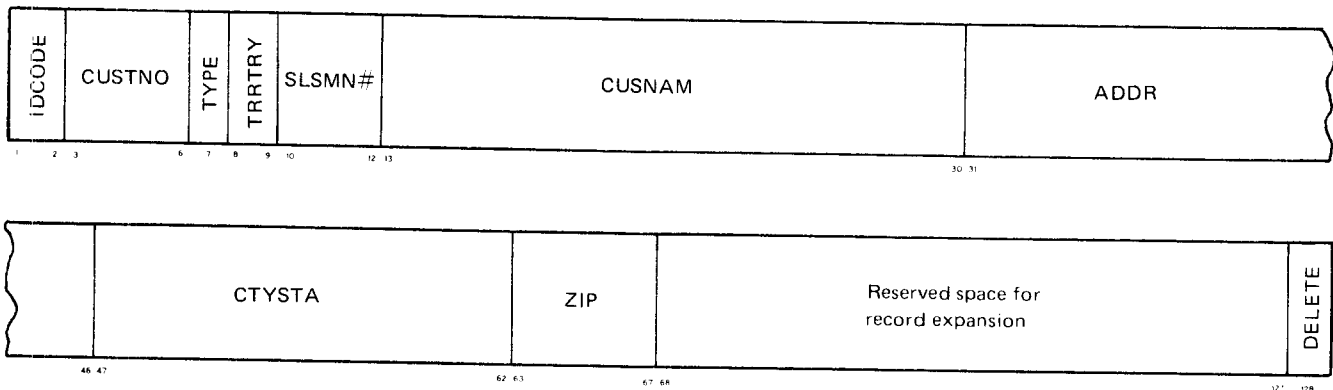


Figure 17. Input Record and Output Record Formats

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
H	1	CUSTOMER	TYPE	NAME												ADDRESS	ZIP CODE	TERRITORY	SALESMAN																																																												
D	3	XXXX	X	XXXXXXXXXXXXXXXXXXXX												XXXXXXXXXXXXXXXXXXXX	XXXXX	XX	XXX																																																												
	4	(CUSTNO)	(TYPE)	(CUSNAM)												(ADDR)	(CITYSTA)	(ZIP)	(TRTRY)	(SLSMN#)																																																											
	5																																																																														
	6																																																																														

CUSTOMER	TYPE	NAME	ADDRESS	ZIP CODE	TERRITORY	SALESMAN	
1637	B	JONES VARIETY	14 S MAIN	BEDROCK, TEX	45412	12	015
4301	B	JIM'S 5 AND 10	1103 FRANKLIN ST	GLENCOE, MN	55336	12	015
3601	D	SCHMIDT HARDWARE	600 1ST ST NW	HILL CITY, MD	21222	02	046

Figure 18. Report of Customer Records

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date \_\_\_\_\_

Program \_\_\_\_\_

Programmer \_\_\_\_\_

Punching Instruction	Graphic								
	Punch								

Page **01**

Program Identification 

75	76	77	78	79	80
----	----	----	----	----	----

Control Card Specifications

Line	Form Type	Core Size to Compile	Object Output Listing Options	Core Size to Execute	Debug	MFCM Stacking Sequence	Input-Shiftings	Output-Shiftings	Inverted Print	360/20 2501 Buffer	Number Of Print Positions	Alternate Collating Sequence	Model 20	Model 20	Refer to the specific System Reference Library manual for actual entries.
01	H														

File Description Specifications

Line	Form Type	Filename	File Type	File Designation	End of File	Sequence	File Format	Block Length	Record Length	L/R	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	Extension Code E/L	Device	Symbolic Device	Labels S/N/E/M	Name of Label Exit	Extent Exit for DAM	Core Index	File Addition/Unordered	Number of Tracks for Cylinder Overflow	Number of Extents	Tape Rewind	File Condition U1-U8
02	F	CUSTCARD	PE	F	96	96																						
03	F	CUSTFILE	CC	F	256	128R																						
04	F	CUSTLIST	0	F	100	100																						

The direct file being created is defined as a chained output file.

Chained output file

Random processing

RPG INPUT SPECIFICATIONS

Date \_\_\_\_\_

Program \_\_\_\_\_

Programmer \_\_\_\_\_

Punching Instruction	Graphic								
	Punch								

Page **02**

Program Identification 

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Filename	Sequence	Number (EN) Blank (O)	Record Identifying Indicator	Record Identification Codes			Field Location		Field Name	Control Level (L-L)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position	
						1	2	3	From	To					Plus	Minus	Zero or Blank		
01	I	CUSTCARDNS	01																
02	I									2	5								
03	I									6	6								
04	I									8	8								
05	I									11	11								
06	I									29	29								
07	I									45	45								
08	I									61	61								
09	I									66	66								
10	I									66	66								
11	I									2	66								
12	I																		

Figure 19 (Part 1 of 2). Creating a Direct Customer File





**IBM** International Business Machines Corporation Form X21-9093 Printed in U.S.A.

### RPG CALCULATION SPECIFICATIONS

Date: \_\_\_\_\_

Programmer: \_\_\_\_\_

Punching Instruction: 

Graphic					
Punch					

Page: 

1	2
03	

Program Identification: 

75	76	77	78	79	80

Line	Form Type	Control Level (LD, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			Not	And	And								Arithmetic	Plus	Minus	
01	C		05			CHANUM	CHAIN	DIRECT								
02	C						SETON						LR			

**IBM** International Business Machines Corporation Form X21-9090 Printed in U.S.A.

### RPG OUTPUT - FORMAT SPECIFICATIONS

Date: \_\_\_\_\_

Programmer: \_\_\_\_\_

Punching Instruction: 

Graphic					
Punch					

Page: 

1	2
04	

Program Identification: 

75	76	77	78	79	80

Line	Form Type	Filename	Type (H/D/T/E)	Space	Skip	Output Indicators			Field Name	Edit Codes	Blank Alter (B)	End Position in Output Record	P = Packed/B = Binary	Sterling Sign Position
						Not	And	And						
01	O	DIRECT	D					05						
02	O								RECORD		96			

Edit Codes

Commas	Zero Balances to Print	No Sign	CH	-	X	Remove Plus Sign
Yes	Yes	1	1	J	Y	Date
Yes	No	2	2	K	Z	Field Edit
No	Yes	3	C	L		Zero Suppress
No	No	4	D	M		

Constant or Edit Word

Figure 20 (Part 2 of 2). Defining a Direct File and Loading Only the First Record

The disk file, which is specified as a chained output file, is cleared to blanks by disk system management after the job begins. The CHANUM field from the input file is used to chain to the corresponding location in the direct file, and the first record is placed in the file. The last record (LR) indicator is then turned on by a SETON operation, forcing the end-of-job condition. The direct file now contains a single record. This job can be immediately followed by one or more jobs which read the remaining records from the MFCU and write out the disk records, using the update function.

After your direct file is defined and cleared to blanks, different steps are required to put records into the file, depending on the method you use to handle synonyms. There are several ways to handle synonyms. Two of the most common methods are:

1. Storing all synonyms in an area of the file set aside for them.
2. Storing synonyms in unused record locations between the records in the file.

If the first method is used, all records can be placed in the direct file in a single job. That job would retrieve and check each record location before it is filled. If the location already contains a record (that is, the record to be written is a synonym), the synonym is stored in the next available location in the part of the file set aside for synonyms. Thus, all home records and synonyms are placed in the file in a single job. A home record is the first synonym record; it is stored in the record location indicated by its relative record number. The rest of the synonyms are then linked together so they can all be found by locating the home record. For more information, see the *IBM System/3 Disk Concepts and Planning Guide*, GC21-7571.

If the second method is used, two jobs are required to place home records and synonyms in the direct file. The first job loads all home records; synonyms are bypassed. The second job loads synonyms in the record locations available between home records. Both jobs are done using the update function to check each record location.

Whatever method you use to handle synonym records, you will have to devise a sequence of jobs similar to those just described. Remember:

1. You load a disk file as a direct file by specifying a chained output file.
2. In order to check for synonyms, you must use the update function. Random update with a direct file is described later in this chapter.

#### **RETRIEVAL OF RECORDS IN A DIRECT FILE**

Record retrieval is used when you want to get information from a record, but do not want to modify the record. You would normally use this method when you want to get data from a record to produce a report. Record retrieval can be done either consecutively, randomly by relative record number, or randomly by ADDRROUT file. For a discussion of record retrieval done randomly by ADDRROUT file, see Chapter 4 in this manual.





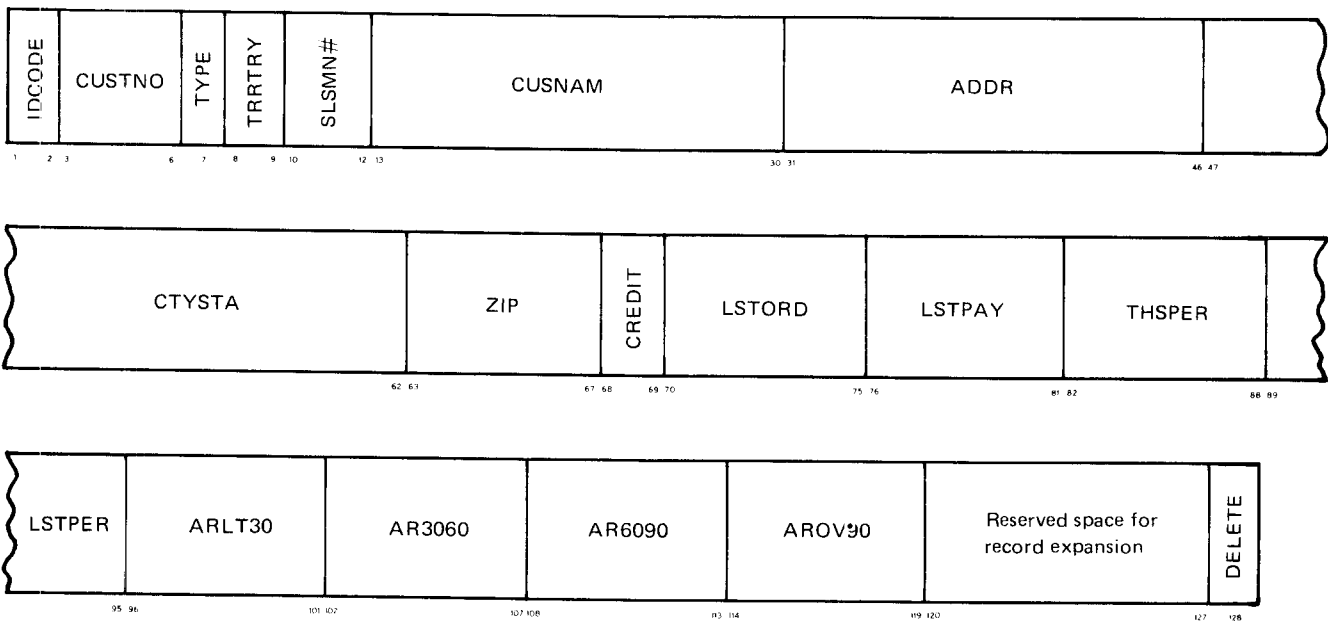
*Example of Consecutive Retrieval*

Suppose you want to process the direct customer file, CUSTFILE, created in the *Example of Creating a Direct File without Synonyms*, to produce a monthly report. This report lists all customers that have had no sales activity during the period. This report is analyzed by sales personnel, who then make follow-up calls. Since all the customer records will be checked and since the file is in sequence by customer number, the report is produced by consecutive processing of the direct file.

The format of the disk records in CUSTFILE is shown in Figure 21. Figure 22 shows a part of the report produced by the consecutive processing job. The report consists of fields selected from CUSTFILE and an accumulated total for accounts receivable (TOTAR).

Figure 23 shows the specification sheets necessary to consecutively retrieve records from CUSTFILE to produce REPORT1, which is a list of recently inactive customers.

Since the direct file probably contains blank record locations and inactive records, a technique is employed on the Input sheet to bypass such records (Figure 23). If a method is not used to bypass unidentified records, the program halts when they are encountered.



Key			
IDCODE	= Identification code	LSTORD	= Last order date
CUSTNO	= Customer number	LSTPAY	= Last pay date
TYPE	= Type	THSPER	= Charges for this period (month)
TRRTRY	= Territory	LSTPER	= Charges for last period (month)
SLSMN#	= Salesman number	ARLT30	= Accounts receivable for less than 30 days
CUSNAM	= Customer name	AR3060	= Accounts receivable for 30 to 60 days
ADDR	= Customer street address	AR6090	= Accounts receivable for 60 to 90 days
CTYSTA	= City and state	AROV90	= Accounts receivable for over 90 days
ZIP	= Zip code	DELETE	= Delete code
CREDIT	= Credit code		

**Figure 21. Disk Record Format for the Direct Customer File**















**IBM** International Business Machines Corporation Form X21-9093 Printed in U.S.A.

### RPG CALCULATION SPECIFICATIONS

Date: \_\_\_\_\_ Program: \_\_\_\_\_ Programmer: \_\_\_\_\_

Page 1 of 2 **03** Program Identification: 75 76 77 78 79 80

Line	Form Type	Control Level (LCL9, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions (Half Adjusted)	Resulting Indicators		Comments
			And	And	And							Arithmetic	Compare	
01	C		07			CSTMER	CHAINCUSTFILE						13	The customer number from the input record is used as the relative record number to chain to the direct file. Indicator 13 will turn on if a record is not found in the direct file.
02	C		07N13			Z-ADDARLT30	TOTAR	72						
03	C		07N13			TOTAR	ADD AR3060	TOTAR						
04	C		07N13			TOTAR	ADD AR6090	TOTAR						
05	C		07N13			TOTAR	ADD AR0V90	TOTAR						
06	C													
07	C													
08	C													

Indicator 13 is used to condition subsequent operations.

**IBM** International Business Machines Corporation Form X21-9092 U.M. 050\* Printed in U.S.A.

### RPG OUTPUT SPECIFICATIONS

Program: \_\_\_\_\_ Date: \_\_\_\_\_ Programmer: \_\_\_\_\_

Page 1 of 2 **04** Program Identification: 75 76 77 78 79 80

Line	Form Type	Filename	Output Indicators				Field Name	Field Length	Constant or Edit Word	Remarks
			And	And	And	And				
01	O	PRINTOUTH	204			1P				} (Other fields - see printed report)
02	O	OR				OF				
03	O									
04	O									} (Other headings - see printed report)
05	O									
06	O									} (Other fields - see printed report)
07	O									
08	O									} (Other fields - see printed report)
09	O									
10	O									} (Other fields - see printed report)
11	O									
12	O									} (Other fields - see printed report)
13	O									
14	O									} (Other fields - see printed report)
15	O									
16	O									} (Other fields - see printed report)
17	O									
18	O									} (Other fields - see printed report)
19	O									
20	O									} (Other fields - see printed report)
21	O									

N13 means that this line will be printed if a record is found in the direct file.

When a record is not found in the direct file, this line is printed.

Figure 26 (Part 2 of 2). Random Retrieval of Inquiry Requests on the Direct Customer File

## MAINTAINING A DIRECT FILE

After a file is created, file maintenance is usually necessary to keep the file current. Three file maintenance functions apply to direct files:

1. Adding records.
2. Tagging records for deletion.
3. Updating records.

### Adding Records

Unlike sequential and indexed files, direct files can have space available between records for new records to be added. Records are added to a direct file by a normal update operation (either consecutive or random processing) as follows:

1. The relative record number is developed for the record to be added.
2. The location is read into main storage.
3. If the location is blank, the new record can be stored.
4. If the location is occupied, and the program can handle synonyms, the new record can be stored as a synonym.

For a discussion of the entries needed to add records consecutively, see *Updating Records, Consecutive Updating of Records* in this chapter. For a discussion of the entries needed to add records randomly, see *Updating Records, Random Updating of Records* in this chapter.

If records must be added but the allotted file space is full, you must increase the total space available for the file. The Disk Copy/Dump program can be used to copy the file into a larger area. For an explanation on how to use this program to copy the file, see the *IBM System/3 Model 10 Disk System Control Programming Reference Manual*, GC21-7512, the *IBM System/3 Model 15 System Control Programming Reference Manual*, GC21-5077, or the *IBM System/3 Model 6 Operation Control Language and Disk Utility Programs Reference Manual*, GC21-7516, depending on the system you have.

### Tagging Records for Deletion

Like sequential and indexed file records, direct file records can be identified for deletion by a delete code. This code is usually a single character at a particular location in the record. When the file is processed, your program must check for the delete code; if the code is present, the record can be bypassed.

Since the record has a delete code, the record location is available for a new record. Either a synonym for a different location can be stored there, or the location can be reused by assigning the relative record number to a new record. If the file contains synonyms, be careful not to delete synonym linkage information when you delete a record and reuse the location.







**IBM** International Business Machines Corporation Form X21-9093  
Printed in U.S.A.

### RPG CALCULATION SPECIFICATIONS

Date \_\_\_\_\_ Program \_\_\_\_\_ Programmer \_\_\_\_\_

Page **03** 1 2 75 76 77 78 79 80  
Program Identification \_\_\_\_\_

Line	Form Type	Control Level (L0, L1, L2, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators		Comments
			And	And	Not							Arithmetic	Plus Minus Zero	
01	C		02			Z-ADD	THSPER	LSTPER						
02	C		02			Z-ADD	0	THSPER						
03	C													

**IBM** International Business Machines Corporation Form X21-9090  
Printed in U.S.A.

### RPG OUTPUT - FORMAT SPECIFICATIONS

Date \_\_\_\_\_ Program \_\_\_\_\_ Programmer \_\_\_\_\_

Page **04** 1 2 75 76 77 78 79 80  
Program Identification \_\_\_\_\_

Line	Form Type	Filename	Space			Skip			Output Indicators			Field Name	Edit Codes	End Position in Output Record	Sterling Sign Position
			Before	After	Before	After	And	And	Not						
01	O	CUSTFILED										THSPER		88	
02	O											LSTPER		95	
03	O														
04	O														
05	O														

Consecutive update of a disk file is done only at detail time.

Commas	Zero Balances to Print	No Sign	CR	-	X	Remove Plus Sign
Yes	Yes	1	A	J	Y	Date
Yes	No	2	K	L	Z	Field Edit
No	Yes	3	C	M		Zero
No	No	4				Suppress

Constant or Edit Word

Figure 27 (Part 2 of 2). Consecutive Updating of Records in the Direct Customer File



*Example of Random Updating*

Each day you want to prepare invoices for customer orders for the file described in the *Example of Consecutive Retrieval*. Information from the invoices is used to update the customer file, CUSTFILE. Since this information is read from records (Figure 28) in an unordered manner, a random update job is required.

The input records contain the date and total amount of the transactions for each customer. New addresses are also on this record when required. As each record is read, the customer number (CUSTMR) is used to chain to the direct file. The amount of the transaction is added to total sales for the period (THSPER) and to the accounts receivable amount (ARLT30). The transaction date is placed in the date of last order field (LSTORD) in the customer record.

If an address change is indicated (an X in column 18 of the input record), the new customer address replaces the old. If a record is not found in CUSTFILE because of an invalid relative record number, the input record is printed, followed by the statement "Above record not found—invalid customer number."

CUSTFILE, described as a chained update file, must be described on both Input and Output-Format sheets because data is read from and written on the file. The specifications are shown in Figure 29.

IDCODE	CUSTMR	DATE	TOCOST	NEWADR	NAMADD
1	2	5 6	11 12	17 18 19	23

Key	
IDCODE	= Identification code
CUSTMR	= Customer number
DATE	= Date
TOCOST	= Total transaction amount
NEWADR	= New address code
NAMADD	= Customer name and new address

Figure 28. Daily Invoicing Record





**IBM** International Business Machines Corporation Form X21-9093 Printed in U.S.A.

### RPG CALCULATION SPECIFICATIONS

Date \_\_\_\_\_ Page 03 of 77  
 Program \_\_\_\_\_ Program Identification \_\_\_\_\_  
 Programmer \_\_\_\_\_

Line	Form Type	Control Level (LD, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			And	And	Not								Arithmetic	Plus	Minus	
01	C		10			CUSTOMR	CHAIN	CUSTFILE								
02	C		10N13			THSPER	ADD	TOCOST	THSPER							
03	C		10N13			ARLT30	ADD	TOCOST	ARLT30							
04	C		10N13			NEWADR	COMP	'X'								
05	C															

**IBM** International Business Machines Corporation Form X21-9090 Printed in U.S.A.

### RPG OUTPUT - FORMAT SPECIFICATIONS

Date \_\_\_\_\_ Page 04 of 77  
 Program \_\_\_\_\_ Program Identification \_\_\_\_\_  
 Programmer \_\_\_\_\_

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Output Indicators			Field Name	Edit Codes	End Position in Output Record	Packed B	Binary	Sterling Sign Position
						And	And	Not						
01	O	CUSTFILED				10N13								
02	O							DATE		75				
03	O							THSPER		88				
04	O							ARLT30		101				
05	O					15		NAMADD		67				
06	O	PRINT	D	1		10 13								
07	O							RECORD		96				
08	O		D	2		10 13								
09	O													
10	O									24	'ABOVE RECORD NOT FOUND--'			
11	O									47	'INVALID CUSTOMER NUMBER'			
12	O													

Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-	X	Remove Plus Sign
Yes	Yes	1	A	J	Y	Date
Yes	No	2	B	K		Field Edit
No	Yes	3	C	L	Z	Zero Suppress
No	No	4	D	M		

Constant or Edit Word

Figure 29 (Part 2 of 2). Random Updating of Records in the Direct Customer File

## Chapter 4. Record Address Files

Record address files are input files that indicate (1) which records are to be read from disk files and (2) the order in which the records are to be read from the disk file. There are two types of record address files:

1. Files containing relative record numbers.
2. Files containing record key limits.

This chapter explains how to use the RPG II language to process these two types of record address files. Sample jobs are used to illustrate the functions.

To understand the sample jobs, a basic knowledge of RPG II is necessary. If you do not fully recall some of the coding used in the sample jobs, you should refer to the *IBM System/3 RPG II Reference Manual, SC21-7504* (for Model 10 Disk System or the Model 15), or the *IBM System/3 Model 6 RPG II Reference Manual, SC21-7517*, depending on the system you are using.

### FILES CONTAINING RELATIVE RECORD NUMBERS (ADDRROUT FILES)

A record address file that contains relative record numbers is called an ADDRROUT file. ADDRROUT files are comprised of 3-byte, binary, relative-record numbers that indicate the relative position (first, twentieth, ninety-ninth, etc.) of records in the file to be processed.

An ADDRROUT file can only be a disk file. All types of file organization (sequential, indexed, or direct) for primary or secondary files can be processed by an ADDRROUT file. When an RPG II program uses an ADDRROUT file to process a file, it reads a relative record number from the ADDRROUT file, then locates and reads the record situated at that relative position in the file being processed. Only records with relative record numbers in the ADDRROUT file are processed.

#### Creating an ADDRROUT File

An ADDRROUT file is created by the Disk Sort program. For an explanation on how to create an ADDRROUT file, see the *IBM System/3 Disk Sort Reference Manual, SC21-7522*.







## FILES CONTAINING RECORD KEY LIMITS

A record address file with record key limits contains the lowest and the highest key fields for a specified section of an indexed file. Record address files containing record key limits can be entered from disk, card, or a printer-keyboard. They are used to process only indexed files. When a section of an indexed file is processed using record key limits, the processing method is known as sequential within limits.

### Creating a File with Record Key Limits

The following rules must be observed when you create a record address file with record key limits:

- Only one record address file can be used for each RPG II program, but the record address file can contain several sets of limits.
- Only one set of limits is allowed on each record in a record address file. Since a set of limits is comprised of two keys, the length of each record in a record address file is twice as long as the length of the record key.
- The low record key must begin in position 1 of the record.
- The high record key must immediately follow the low record key. No spaces are allowed between the two keys. If the key field were four bytes long and the low record key were 2000 and the high record key 3000, the record would look like this:

20003000

- Each record key can be from 1 through 29 characters long.
- An alphameric record key can contain blanks.
- The length of the keys must equal the length of the key field in the indexed file. To make the length of the keys equal, leading zeros may have to be placed in a numeric record key or blanks in an alphameric record key. For example, if the low record key were three positions (200), the high record key four positions (2999), and the length of the key field in the indexed file four positions, a zero must be placed before the 200 to make it a 4-position number. The record would look like this:

02002999

Each key length must also equal the key field length you specify in columns 29 and 30 of the File Description sheet.

- The same set of limits can appear on more than one record in a record address file. Therefore, records within a set of limits can be processed as many times as needed.
- The two record keys in a set of limits can be identical. For example, both the low and high record keys can be 2999. In this case, only one record (2999) will be processed.





All records in a file must be the same length. Thus, column 19 contains an F to indicate that the record length is fixed.

A number equal to or a multiple of the disk record length must be entered in columns 20 through 23. This entry determines the size of the input/output area allocated by RPG II. For an explanation on block length calculation, see the *IBM System/3 Disk Concepts and Planning Guide*, GC21-7571. If you want block length calculated for you by RPG II, assign a block length equal to the record length. By blocking disk records you can increase the input/output efficiency of your program by reducing the number of accesses. You must be sure, however, that enough main storage is available for your input/output area.

Columns 24 through 27 must contain the length of the disk record. Key length, entered in columns 29 and 30, is the same length for all records in a file. The maximum record key length is 29 characters.

Whenever a disk file is described, DISK (Models 6, 10, and 15), DISK40 (Model 12 or 15), or DISK45 (Model 10, 12, or 15) is required in the Device columns. Since additional information is given on the Extension sheet, column 39 must contain an E.

### Extension Specifications

To process a file using a record address file, two entries must be made on the Extension sheet:

International Business Machines Corporation

**RPG EXTENSION AND LINE COUNTER SPECIFICATIONS**

Punching Instruction	Graphic Punch
----------------------	---------------

**Extension Specifications**

Number of the Changing File		To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	S C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 * . - /	Table or Array Name (Automatic Format)	Length of Entry	S C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 * . - /																																																																															
From Filename	Number of the Changing File																																																																																								
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	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00

Columns 11 through 18 must contain the name of the record address file. The name must be the same as the record address file described on the File Description sheet.

Columns 19 through 26 must contain the name of the file to be processed. This name must be the same filename defined on the File Description sheet. This entry indicates that the file is to be processed by the record address file coded in columns 11 through 18.

### Example of Processing Sequentially within Limits

You have a master customer file on disk consisting of 128-character records. The file is organized by customer number within customer class. Customers are separated into such classes as wholesalers or retailers. Together the customer number and class form a composite customer account number (key) in the form: ccnnnnn.

The customer class is cc and the customer number is nnnnn. Customer classes begin at 01 and are in ascending order. Within each customer class, customer numbers range from 00000-99999.

You must prepare separate reports for each customer class for sales analysis purposes. A record address file can be used to supply the particular class categories and customer number ranges as shown in Figure 31. The key in each disk record begins in position 2, and the record address file is loaded in MFCU1. Figure 32 shows the necessary File Description and Extension entries for this job.

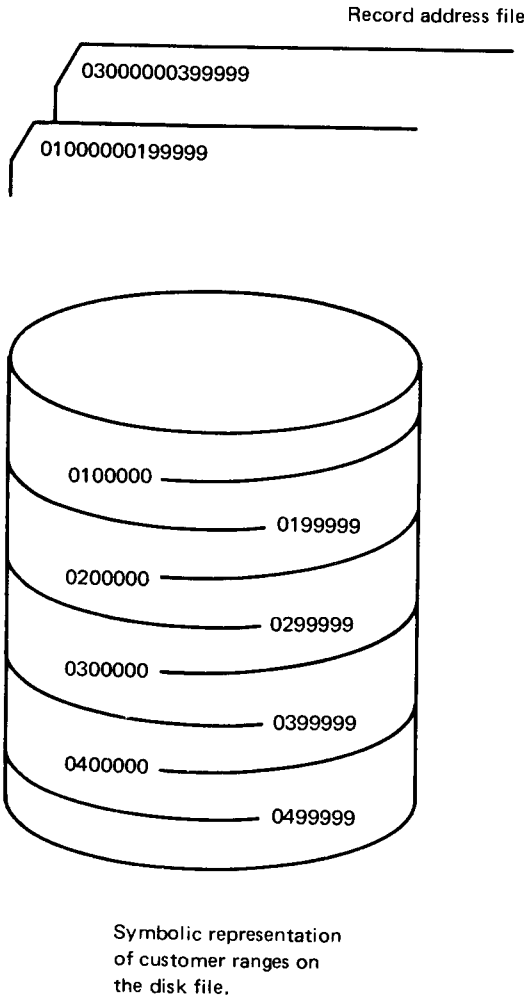


Figure 31. Files for Processing Sequentially within Limits





- adding records
  - direct file 70
  - indexed file
    - description 39
    - randomly by key using chaining 37
    - randomly by key without chaining 43
    - sequentially by key 44
  - sequential file
    - adding at the end of records in the file 7
    - description 7
    - merging between records in the file 12
- ADDRROUT file
  - creating 79
  - description 79
  - processing by 80

- building a disk file v
- CHAIN operation code
  - adding records to an indexed file 37
  - creating a direct file 53
- consecutive processing
  - direct file 51, 61
  - indexed file 50
  - sequential file 1, 16
- copy/dump (see reorganizing files)
- DELETE parameter (disk copy/dump) 22

- creating files
  - ADDRROUT file 79
  - direct file
    - description 51
    - with synonyms 58
    - without synonyms 53
  - indexed file
    - description 26
    - ordered file 27
    - unordered file 28
  - record key limits file 83
  - sequential file 1

- direct files
  - adding records 70
  - creating
    - description 51
    - with synonyms 58
    - without synonyms 53
  - description 51
  - record retrieval
    - consecutive 61
    - description 60
    - random 66

- tagging records for deletion 70
- updating records
  - consecutive 72
  - description 71
  - random 75
- disk copy/dump (see reorganizing files)

- indexed files
  - adding records
    - description 39
    - randomly by key using chaining 37
    - randomly by key without chaining 43
    - sequentially by key 44
  - consecutive processing 50
  - creating
    - description 26
    - ordered file 27
    - unordered file 28
  - description 26
  - random processing
    - by key 33, 37, 43
    - by relative record number 50
  - reorganizing a file 49
  - tagging records for deletion 49
  - updating records
    - description 32
    - randomly by key 33
    - sequentially by key 33
    - sequentially within limits 49, 84
- inquiry request 67

- master file vi
- merging records between records in a sequential file 12
- multivolume disk files (see note in preface)
- OMIT parameter (disk copy/dump) 22

- ordered indexed file
  - creating 27
  - description 26

- processing files
  - by ADDRROUT file 80
  - sequentially within limits 84

- random processing
  - direct files (see direct files)
  - indexed files (see indexed files)
- record address files
  - ADDRROUT file
    - creating 79
    - description 79
    - processing by 80
  - description 79
  - with record key limits
    - creating 83
    - description 83
    - processing 84
- record key limits file
  - creating 83
  - description 83
  - processing 84
- record retrieval for direct files
  - consecutive 61
  - description 60
  - random 66
- relative record number
  - direct files 52
  - indexed files 50
- reorganizing files
  - indexed file 49
  - sequential file 22
- retrieval of records (see record retrieval for direct files)

- sequential files
  - adding records
    - adding at the end of records in the file 7
    - description 7
    - merging between records in the file 12
  - creating 1
  - description 1
  - reorganizing a file 22
  - tagging records for deletion 16
  - updating records 16
- sequential within limits processing 84
- synonyms (direct files) 58, 53

- tagging records for deletion
  - direct file 6
  - indexed file 49
  - sequential file 16
- transaction file v

- unordered indexed file
  - creating 28
  - description 26
- updating records
  - direct file
    - consecutive 72
    - description 71
    - random 75
  - indexed file
    - description 32
    - randomly by key 33
    - sequentially by key 33
    - sequentially within limits 49, 84
  - sequential file 16



# Technical Newsletter

This Newsletter No.	GN21-5427
Date	24 September 1976
Base Publication No.	GC21-7566-1
File No.	S3-38
Previous Newsletters	None

## IBM System/3 RPG II Disk File Processing Programmer's Guide

© IBM Corp. 1971, 1974

This technical newsletter, a part of version 05, modification 00, of IBM System/3 Model 15 RPG II, applies to the current versions and modifications of RPG II for all System/3 models, and to subsequent versions and modifications unless specifically altered. This TNL provides replacement pages for the subject publication. Pages to be inserted and/or removed are:

v through viii  
1, 2  
27, 28  
37, 38  
43, 44  
51, 52  
81, 82  
85, 86

Changes to text and illustrations are indicated by a vertical line at the left of the change.

### Summary of Amendments

- Changes effected by 5704-RG2 have been added.
- Information about 5448 Disk Storage Drive has been added.

*Note:* Please file this cover letter at the back of the manual to provide a record of changes.

IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901

© IBM Corp. 1976

Printed in U.S.A.







This Newsletter No. GN21-5625

Date 30 June 1978

Base Publication No. GC21-7566-1

File No. S3-28

Previous Newsletters GN21-5427

## IBM System/3 RPG II Disk File Processing Programmer's Guide

© IBM Corp. 1971, 1974

This technical newsletter applies to the current versions of RPG II for all System/3 models, and to subsequent versions and modifications unless specifically altered. This TNL provides replacement pages for the subject publication. Pages to be inserted and/or removed are:

Title Page, Edition Notice

iii, iv

29, 30

33, 34

34.1, 34.2 (added to accommodate moved text)

35, 36 (text rearranged)

37 through 40

Changes to text and illustrations are indicated by a vertical line at the left of the change.

### Summary of Amendments

Miscellaneous technical changes

*Note:* Please file this cover letter at the back of the manual to provide a record of changes.



# READER'S COMMENT FORM

IBM System/3  
RPG II Disk File Processing  
Programmer's Guide

GC21-7566-1

## YOUR COMMENTS, PLEASE . . .

Your comments assist us in improving the usefulness of our publications; they are an important part of the input used in preparing updates to the publications. All comments and suggestions become the property of IBM.

Please do not use this form for technical questions about the system or for requests for additional publications; this only delays the response. Instead, direct your inquiries or requests to your IBM representative or to the IBM branch office serving your locality.

Corrections or clarifications needed:

<i>Page</i>	<i>Comment</i>
-------------	----------------

Please include your name and address in the space below if you wish a reply.

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

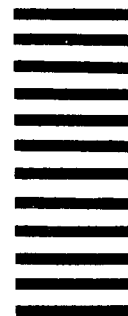
Cut Along Line

Fold

Fold

FIRST CLASS  
PERMIT NO. 387  
ROCHESTER, MINN.

**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

IBM Sy ; RFG II (File No. S3-28) Pr

Fold

Fold

rd in U.S.A. GC21-7566-1



**International Business Machines Corporation**  
General Systems Division  
875 Johnson Ferry Road N. E.  
Atlanta, Ga. 30342  
(USA Only)

**IBM World Trade Corporation**  
821 United Nations Plaza, New York, New York 10017  
(International)





International Business Machines Corporation  
General Systems Division  
875 Johnson Ferry Road N. E.  
Atlanta, Ga. 30342  
(USA Only)

IBM World Trade Corporation  
821 United Nations Plaza, New York, New York 10017  
(International)