IBM

Concepts of System/32

SCP, Utilities, RPG II for the Experienced User

SYSTEM/32

GB 30-0001-0

CONCEPTS OF SYSTEM/32 SCP, UTILITIES, RPG II FOR THE EXPERIENCED USER

First Edition (January 1975)

PREFACE

This manual discusses some of the differences between the IBM System/32 and the IBM System/3 to help the experienced System/3 customer and IBM Systems Engineers to rapidly understand the System/32. The reader should be familiar with System/3 OCL, Utilities, Sort and RPG II.

The manual consists of three sections:

Section 1.0 explains the System/32 System Control Programming OCL changes, the altered functions of the system utility programs, new disk usage concepts, and some system operation differences.

Section 2.0 discusses the functions offered by the System/32 Utilities Program Product (5275-UTI) for managing data files, entering source programs and procedures, and sorting.

Section 3.0 is written for the person who has coded, compiled and tested programs using the RPG II language on an IBM System/3 Disk System. It discusses the differences in the System/32 RPG II language. Four subsections are used to briefly review RPG II, point out specific differences that are unique to the System/32, identify System/3 RPG II coding that is not available in the System/32 RPG II language, and discuss the Interactive Data Entry function of the System/32 RPG II language.

A two-part Appendix is included. Part 3A presents a brief description of the Auto Report feature, and Part 3B is a guideline to System/32 operation by programmers who will compile and test their own programs.

For more detailed information on the discussed topics, refer to:

Introduction to System/32	GC21-7582
System/32 SCP Reference Manual	GC21-7593
System/32 Operator's Guide	GC21-7591
System/32 Messages Guide – System	GC21-7592
System/32 Utilities Program Product Reference Manual – Data File Utility	SC21-7600
System/32 Utilities Program Product Reference Manual – Source Entry Utility	SC21-7605
System/32 Utilities Program Product Reference Manual – Sort	SC21-7633
System/32 Messages Guide – Utility Program Product	GC21-7618
System/32 RPG II Reference Manual	GC21-7595
System/32 Messages Guide – RPG II	SC21-7617

Note: The following items will be available with Release 2 of System/32 and are described in this manual for planning purposes only.

- A disk storage capacity of 9,154,560 characters of information.
- Main storage capacities of 24,576 and 32,768 characters of information.
- The Serial Print Capability
- Eject/No Eject Parameter of // LOG OCL statement

CONTENTS

SECTION 1.0 – SYSTEM/32 SYSTEM CONTROL PROGRAMMING

Although the support provided by the System/32 SCP is quite similar to that provided by the System/3 SCP, there are a number of areas where the System/32 support differs. This is due to the different architecture of the system, new operator interface or enhanced functions. These areas include:

- Operator Control Language (OCL)
- Utility Program Control Statements
- System Operation
- Disk Usage

1.1 – Operator Control Language (OCL)

The System/32 OCL is derived from the System/3 Model 10 OCL and is compatible in many of the statements. However, some System/3 statements have been eliminated; some have been modified and some statements have been added for System/32.

System/3 – Only Statements

The following statements appear in System/3 OCL but are not a part of System/32 OCL. These statements will be diagnosed as an optional form of the // INCLUDE statement by the System/32 SCP. The word following the // will be considered a procedure name which may or may not be in the library.

// CALL

This statement has been replaced by the INCLUDE statement in System/32 OCL.

// HALT, // NOHALT

These two statements and the corresponding option are not applicable to System/32. A nested procedure on System/32 is the only method by which multiple jobs may be executed without operator intervention.

// READER

This statement is not supported by System/32 since all OCL statements must either be entered from the keyboard or be contained within a stored procedure in the disk library.

// PARTITION, // LOCKOUT

The System/32 does not require these statements since there is no dual programming capability on the system as on System/3.

// PUNCH

The System/32 does not support a real or simulated punch function.

/& This end-of-job statement is used on System/3 to separate OCL and data statements into logical jobs and is not required on System/32 since all OCL must come from the keyboard or a stored procedure.

Modified Statements

The following System/3 OCL statements are also supported on the System/32 but some of the keyword parameters have been changed or eliminated. Where a keyword is not used by System/32, that parameter must not be entered since it will be diagnosed as invalid and the entire statement will be dropped.

// LOAD

The UNIT parameter is not required by System/32 since there is only one disk unit.

The // LOAD * form of the statement is not valid on System/32.

// DATE

A third format of the date value is supported by System/32. Date may be entered in the format, YEAR/MONTH/DAY using the same rules of punctuation which apply to the previous formats (D/M/Y, M/D/Y).

// SWITCH

This statement may appear anywhere in the sequence of OCL statements for a job or within a procedure on System/32.

Multiple // SWITCH statements may appear in the OCL stream for a single job on System/32. Only one of those statements may appear between the // LOAD and the // RUN statement, however.

In System/3 OCL, only one SWITCH statement is permitted and must appear between the LOAD and RUN statements.

// LOG

On System/32, this statement cannot be used to turn SYSLOG on or off since SYSLOG is always 'on'.

A parameter is provided on System/32 which permits the SYSLOG output to be assigned to the Display Screen only (CRT) or to the Display Screen and the printer (PRINTER). Regardless of how SYSLOG information is displayed, it is stored in a disk history file for later reference.

A second parameter (EJECT/NO EJECT) is supported which permits printer output to be ejected (skip to line one) of left positioned as is at end-of-job. Release 1 provides automatic EJECT at EOJ.

// IMAGE

No UNIT parameter is supported on System/32 since there is a single disk library.

The maximum number of characters which may be entered for the print image on System/32 is 192 or 384 if hexadecimal characters are entered.

// COMPILE

No UNIT or OBJECT parameters are supported on System/32 since there is a single disk library.

// FILE

The PACK parameter applies only to diskette files on System/32.

The UNIT parameter is optional on System/32. A default to UNIT-F1 will be used if the parameter is omitted. The value I1 is used to indicate a diskette file. The values R1, R2, F2 are not supported.

On System/32 the disk space for a file may be specified in RECORDS (as on System/3) or BLOCKS. A block consists of 10 sectors (2560 bytes.) The System/3 keyword, TRACKS, is not supported on System/32.

Corresponding to the above space allocation, System/32 file LOCATION specification is in terms of the beginning BLOCK number. The disk space on System/32 is divided into 10-sector blocks beginning with Block 1 at Cylinder 1, Track 1, Sector 9.

NOTE: See the discussion related to use of the LOCATION parameter in the following section.

System/32 supports RETAIN values of P,T,S, for the disk (no RETAIN-A or RETAIN-nnn for diskette files. The value of nnn may be 000 to 999. Values 000-998 are added to the current system date to arrive at an expiration date to place in the diskette file label. The value 999 is used to allocate a permanent diskette file.

New System/32 Statements

The following statements have been added to System/32 OCL to support new function and generally enhance the stored procedure power and flexibility.

// SYSLIST

A number of the System/32 utilities and the RPG II compiler use the system printer function called SYSLIST. This new OCL statement has been added to permit assigning the SYSLIST output to the printer, display or preventing the output from appearing on either device.

// MEMBER

System/32 uses a set of text messages stored in the disk library for most of the operator communication messages and page headings displayed and printed by the SCP, utilities, RPG compiler, OCL message statements and user program messages. The MEMBER statement is provided to permit selection of the particular set of text messages to be used by user OCL statements and programs at a given point in the processing of a job stream.

// Message Number or 'message text'

System/32 OCL includes this statement to permit OCL procedures to contain operator prompting or information messages. The message may be contained within the statement itself or may be retrieved from a user message member which was previously identified to the SCP using the MEMBER statement.

// INCLUDE

The System/32 INCLUDE statement performs the function for the System/3 CALL statement plus added functions.

- Up to 10 positional parameters may be passed to the invoked OCL procedure statements.
- Any combination of OCL statements may be included in the invoked procedure (not restricted to a LOAD-RUN set).
- The // INCLUDE may be omitted, resulting in single word "commands."
- Up to 16 levels of nested INCLUDE statements are supported.

Conditional Clauses

Two conditional expressions may be used in procedures along with System/32 OCL statements to cause the OCL statement to be accepted or bypassed.

- The IF expression allows testing of data file or library member existence, switch setting (user switches can now be set on or off by RPG) or comparison of two values. IF may take the forms IF or IFT (if true), or IFF (if false).

- The ELSE expression is used to condition the processing of an OCL statement based on the results of a preceding IF test.

These two clauses used in conjunction with the new INCLUDE statement, operator communication and parameter testing/passing/default capabilities produce a very useful procedure "language."

1.2 – System Utilities, Utility Control Statements

In addition to the OCL statements described above, the utility program control statements are frequently contained in OCL procedures. While many of the System /32 utility programs are similar in function to those in System/3 support, almost all have some new or altered functions. A partial list of the altered function for some of the commonly used utilities follows. The changes to the control statements are not identified here but may be found in the System/32 SCP Reference Manual, SC21-7593.

General

Single statement commands have been constructed for the common System/32 utility functions and are distributed with the SCP. They may be entered from the keyboard or coded within OCL procedures to avoid keying or coding more extensive OCL statements. When the user desires a utility function which is not included in the commands provided, he should reference the utility program description to determine if the desired function is available by coding detailed OCL and utility control statements.

NOTE: The reader is reminded that a command is simply an INCLUDE statement in the optional format which omits the // INCLUDE portion.

$COPY Utility

Added function supports copy to and from the diskette.

The SELECT capability and the OMIT capability apply to a disk-to-disk, disk-to-print, disk-to-diskette, and diskette-to-disk copy.

Packed numeric keys are supported in the selection specifications.

Copy and print combination is not supported on System/32.

$MAINT Utility (Librarian)

System/32 supports library-to-file and file-to-library movement for disk and diskette files.

Copy library-to-library with NEWNAME is supported.

Copy to/from diskette is supported

Library create, delete, reorganize are not supported.

$DELET Utility

Processes diskette and disk files

The 'Scratch' function is treated the same as 'Remove' for the disk (since there is no RETAIN-A support). 'Scratch' for diskette changes the expiration date to the current date.

The delete VTOC (all files) is not available through the DELETE command but may be accomplished using OCL statements and a file name of LABEL-ALL.

$INIT Utility

Only the diskettes may be initialized.

$ALT Utility

This utility does not exist in System/32 support. Alternate sectors are automatically assigned by the SCP upon detection of a permanent disk error (disk only). The program being executed at the time the failure is detected is cancelled; the alternate sector assigned; an attempt to move the data from the error sector is made and then a SYSLOG message is displayed.

$BUILD Utility

This utility must be executed after an alternate sector has been assigned to cause the alternate sector to become useable by normal disk data management. The utility permits display of the data so that any possible errors may be corrected.

1.3 – Disk Usage

There are several new concepts employed in the usage of the System/32 disk even though the disk data management is essentially identical to System/3 disk data management. The major areas which require consideration are discussed in the following points.

Fixed Disk

Since the disk cannot be removed from the System/32, provision must be made for the following:

- Backup – protection of disk file data by copying the data to another area of disk or to diskettes for offline storage.

- Capacity – When all the user's data will not fit on the internal disk at one time, loading and dumping some files from/to diskettes must be incorporated into the planned job stream.

- Offline files – the System/32 has some support of offline multivolume files' as found in System/3 support. This support is limited to consecutive files (input, output, update) and is accomplished using diskette resident files copied temporarily to internal disk for processing.

Single Library

On System/32 all SCP, Program Product and user programs and procedures are stored within a single library area with a single directory. The directory entries are sorted into sequence by type and name. The modules (members) themselves, however, are stored on disk in the sequence in which they were copied to the disk without regard to type or any other criteria.

Also, when new members are added to the library they are placed after the current 'last member'. Space ('holes') in the library caused by deleting certain members is not reused until a reorganization is accomplished by dumping the entire library to diskette and reloading.

This organization must be considered when programs or procedures are to be loaded to the disk on a temporary basis, then removed to make room for another group of programs. For example, a compiler may be temporarily loaded online for program maintenance and then removed. If a new object program (load member) is created in the above example and the compiler is then deleted, the space occupied by the compiler is not available.

In this example, the space freed by deleting the compiler could be made available by the following sequence of jobs.

- Load compiler

- Compile

- Copy object program to a data file

- Delete compiler members

- Delete new object program member

- Copy object from data file to library

- Delete data file

The size of the library may be changed whenever it is reloaded from diskettes or by use of the ALLOCATE statement with $MAINT. However, the size of the library directory can only be changed during the reload from diskettes.

Disk File Allocation

All disk files on the System/32 internal disk are allocated in units of 'BLOCKS' (2560 bytes, 10 sectors) and the index area of an indexed file will be suballocated in units of BLOCKS.

When a new file is allocated, it will be positioned as close to the library as possible (lowest cylinder numbers), unless LOCATION is specified regardless of whether the file is identified as permanent, temporary or scratch. This means that the library cannot be enlarged unless the data file (s) adjacent to the library are first deleted.

To help preserve the disk organization described above, a $PACK utility program (COMPRESS command) is provided which will remove any free disk space between existing files, caused by file deletion or a reduced library size, by shifting files toward the library and thus optimizing the utilization of the internal disk.

It is important to note that use of the OCL 'LOCATION' parameter (to explicitly position a disk file) is not consistent with the file allocation logic of System/32, which is to let the system locate the files. The parameter is available but must be used only when essential to a unique application requirement. Some of the utility commands (for example, RESTORE, COMPRESS) must not be used if file location is critical. The commands do not support LOCATION specification and files will be automatically relocated on the disk.

A function of System/3 SORT is the capability to write the output file over (same location) the input file. This requires the specification of LOCATION in the OCL for the output file. Since LOCATION specification is not desirable on System/32 it may be necessary to use an alternate SORT file allocation technique.

All System/32 disk addresses are specified as relative selector number. This means that the address is given as sector number relative to the beginning of this data file. Thus, a data file can be shifted on the disk without affecting the program or index disk addresses.

1.4 – System Operation

Operation of the System/32 most closely resembles operation of the System/3 Model 6, with the following differences.

OCL

The System/3 Model 6 uses a conversational scheduler to prompt the operator to complete OCL statements. The System/32 input is either:

1. Actual OCL statements
2. Commands (The name of a stored procedure).

The primary mode of operation of System/32 is expected to be the latter of these, resulting in a very easy-to-key stream of operator commands to the system. The procedure capability has been enhanced so that any prompting of the operator which is required can be coded as a part of the procedure.

Display

The display screen is always used on System/32 as the SYSIN display and SYSLOG for immediate reference as well as to recall and display previous commands and messages from the history file.

This history file is fixed in size (39 sectors) and once filled, begins overlaying the oldest entries with new commands, OCL and SYSLOG messages. If a printed copy of the history is required, the operator should be instructed to print the information (using a utility command) at convenient breaks in the daily processing stream.

Diskettes

Since the System/32 disk files which are backed-up or kept offline must be read from and written to diskettes, many diskettes will be handled by the operator each day. It is essential then, that the operator have a good means to access, store, label and control numerous diskettes. The diskettes are also useable on other devices, so the control must include a method of protecting system diskettes from key entry use, etc.

SECTION 2.0 – SYSTEM/32 UTILITIES PROGRAM PRODUCT (5275-UT1)

The following utilities make up the System/32 Utilities Program Product:

- Data File Utility (DFU)
- Sort
- Source Entry Utility (SEU)

Neither the Data File Utility nor the Source Entry Utility has a functionally equivalent System/3 Utility for comparison. This discussion is intended only to acquaint the reader with the highlights of each of these utilities. The Sort program is, however, very similar in function to the System/3 product and is treated as such.

2.1 – Data File Utility (DFU)

The following data base management functions are provided as part of DFU:

- Data File Creation and Maintenance
- Data File Inquiry
- Data File List

The above functions take advantage of cataloged RPG II File Description and Input Specifications. To use any of the functions, the user needs to know only the name of the file and the name of the cataloged RPG II specifications. The functions will prompt the user for all other information needed to tailor the job to the user's task. Field names are included with the prompts to aid the user in selecting the data fields to be used.

DFU will process record sizes up to 256 bytes. Data processed in a single record type is limited to a 132 byte printer width, including spacing and column headings.

Data File Creation and Maintenance

This function provides the necessary capabilities to create and maintain data files in a data base. Maximum use is made of the display to prompt the operator, by field name, for the data to be entered. For update, the data currently in the field is displayed. This function operates only with indexed files. Highlights include:

- Formatted Report for an Audit Trail – The user can choose to print all records being entered. All records being modified or deleted will always print. The listing will include page headings (including date and page number) and column headings.

- Auto Duplication of Fields – Through the use of an Auto Duplication indicator, fields may be automatically duplicated from one record to the next. In addition, with the duplication function key, data can be duplicated a character or field at a time, between records.
- Control Totals – A field may be totaled for purposes of checking that the correct data has been entered. A final total will be printed. By using a command key batch totals may also be printed.
- Generated Keys – The user may optionally specify that DFU generate record keys.
- Field Prompting – DFU will prompt the operator for the field to be entered by displaying the designated column heading for the field on the display screen.
- Modulus 10 and 11 Self-Check Digit: The user may optionally choose to use self-check digit for a field and DFU will check to ensure that the correct data was entered.

Data File Inquiry

This function provides the necessary capability to allow inquiry into a file. Any indexed file may be accessed. The current status of the information in the file is displayed.
Highlights include:

- Retrieval by Record Key – The operator may retrieve a record from a file by entering the key of the record.
- The capability to roll forward in key sequence through the file. The operator may roll forward through the file to the record desired through use of a function key.
- The capability to roll backward in key sequence through the file. The operator may roll backward through the file to the record desired through use of a function key.
- The ability to print a record – The operator may print a desired record by using the DFU print command key. The desired record(s) will print complete with page and column headings describing the data printed.

Data File List

This function provides the capability to list and summarize selected information from any indexed or sequential file in the data base. The utility is very useful for obtaining one-time reports and for creating recurring management reports. Highlights include:

- Page headings including date and page number – Page headings print on every page of the listing. The system date, the page number, and the title of the report constitute the page heading.

- Column Headings – Column headings print on every page of the report describing the data being printed. For a record list the appropriate column headings will also be printed every time the record type changes.

- Edited Data Fields – All numeric fields are printed with leading zeroes suppressed, the appropriate decimal point and a following minus sign for negative fields.

- Column Totals – For numeric fields selected to be accumulated, the totals will be printed in the same column as the data field. Up to three levels of subtotals may be printed plus the final totals for the report. Up to five numeric fields may be accumulated in one report.

- Selection Based on Record Codes – Records may be selected for inclusion based upon the record types described in the RPG II Input Specifications for the file.

- Ascending or Descending Sort – The file may be sorted prior to the report being printed. Up to three fields may be sorted. Ascending or descending sequence may be specified for each of the sort fields.

- Summary list with totals only – Several listing options are available:

  1. A report may be printed with all the detail records plus column totals.

  2. The report may contain only the column totals along with the pertinent identification data (group print with group indicate).

  3. The report may be simply a listing of the file where the column headings change for every record type, with control totals being optionally taken.

2.2 – Sort

The System/32 SORT Utility provides the function and capability of the proven System/3 SORT. Highlights include:

- Selection Based on Field Contents – The contents of a field in a record may be compared against another field in the record or a constant, with the records to be sorted being based upon the results of the compare.

- Support of Any Disk Data File – Any file type (indexed, direct, sequential) may be sorted using the SORT Utility. The output is always a sequential data file.

- ADDROUT (Tag) Sort – Work space required for sorting may be reduced by specifying a Tag Sort. Only the control fields and a record address are carried through the sort. The output of the sort is a file containing a string of addresses of the file being sorted. This ADDROUT file may then be read into a processing program and used to retrieve the data file.

- Summary Sort – A summary sort provides the ability to reduce the number of records being sorted and written to the output file. Records with like control fields are brought together with designated numeric fields summed into a single summary record.

- Automatic Work File Allocation – SORT will automatically allocate disk space for a work file if none has been specified through OCL.

- Ascending or Descending Sequence – Sort fields may be sorted in either ascending or descending sequence.

2.3 – Source Entry Utility (SEU)

SEU can be used to create and maintain OCL procedures, RPG II source statements and Sort source statements. Highlights include:

- Sort Format Descriptions – Format descriptions are distributed with SEU which describe Sort source statements and can be used as an aid in entering those statements correctly.

- RPG II and Auto Report Format Descriptions – Format descriptions are distributed with SEU which describe RPG II and Auto Report source statements correctly.

- RPG II and Auto Report Syntax Checking – Optionally, SEU will perform a diagnostic check on RPG II and Auto Report Source Statements as they are entered. The check performed is sufficient to determine that the statement entered contains no syntax errors.

- Resequence Statements – Optionally, at EOJ the operator can choose to have the statements in the member being created or maintained sequenced, and can choose where that sequence number should be stored within the statements in the library.

- Move Statements – The operator can choose to move statements from one spot in the source or procedure member to another spot in the same member.

- Include Statements – Statements from a source (procedure) member in the library may be included in the member being worked on.

- Delete Statements — Statements may be deleted from the source or procedure member.

- Statement Insertion — Up to 99 statements may be inserted between two statements which already exist in a source or procedure member.

- Rolling Forward or Backward — Through use of the function keys, the operator may roll either forward or backward through the source or procedure member until the desired statement is reached.

- Display Screen Usage — The display screen is used to show the entire record being entered or updated. As data is keyed, it appears on the display screen.

SECTION 3.0: RPG II SELF – STUDY COURSE

This section contains a self-study course for the experienced System/3 RPG II programmer. You should be able to completely read and understand its contents within 6 to 10 hours.

A self test follows each section. Answers are provided so that you can check your progress. If you want or need more System/32 RPG II information, you can refer to System/32 RPG II Reference Manual SC21-7595.

Upon completion of this self-study section, you should be able to do three things:

1. Describe, on RPG II specifications sheets, those entries that are exclusively available in the System/32 RPG II language.

2. Make a list of the categories of System/3 Disk System RPG II specifications that have been omitted from the System/32 RPG II language, and

3. Describe, in general terms, the RPG II coding that is used for System/32 Interactive Data Entry.

As you solve problems using the System/32 RPG II language, you will find it convenient to know more about the Auto Report feature of RPG II and the Source Entry Utility (SEU) program. Other items to investigate are the Data File Utility (DFU) and sort programs.

3.1 – "RPG II Review"

The RPG II language is used to describe jobs in terms of files, records, fields and calculations needed to produce desired output from available input.

Among its uses we can include:

1. The creation of reports and other output files,
2. The updating of existing disk files,
3. The use of tables and arrays of data

Coding of specifications is done on up to six different coding forms that contain eight fixed-form types of specifications. When coding is completed, the source program is compiled into an object program that has a fixed program logic. The fixed logic can be somewhat modified by special coding operations such as EXCPT, FORCE, CHAIN, READ, and SETLL.

When generated, the program includes the steps necessary to open, get, put and close files with little coding on the part of the programmer.

Key to the successful use of RPG II lies in the proper use of indicators to identify records, control calculations and produce output records in proper order. There are up to 9 levels of control, 9 fields for matching records and 99 record type identifiers or field condition testers. Other special indicators include 1P for first page control, OA-OG and OV for page overflow control, H1-H9 to halt processing when undesired results or situations occur, and U1-U8 for external file control.

Convenience items include access to the system date by using UDATE, UDAY UMONTH or UYEAR, output record numbering using PAGE, edit codes for numeric fields, constants and literals, AND and OR combinations, and subroutines.

Essentially, an object program will read, process and produce output for one record at a time. Group indication and group printing may be used as needed. When two or more levels of control are specified, a change in a higher level control field automatically forces a change in every lesser level. Output associated with these control levels is produced in order, starting with records controlled at the lowest level. Calculations performed for each record type are detail-time calculations and total time calculations conditioned by L0, while those occuring for a control group are called total-time calculations.

Auto Report is a feature available to RPG II users. Auto Report coding entries are made on RPG II forms and then an RPG II source program is generated. The RPG II source program may be modified by additional coding prior to final compilation if desired. Auto Report provides for centered report page titles and columnar field headings. It is also used to copy pre-defined RPG II statements from the library so that they may be combined with other coded entries prior to compilation. (Note: Appendix 3A in this section provides basic Auto Report details if you are interested).

3.2 – Self-Test

Try to answer these questions **without** reference to the information in Section 1 or the RPG II Reference Manual. Write your answers on a piece of scratch paper. When you have completed all of the questions, turn the page to check your answers against those in the book.

1. Where are files defined? Where are their records described?

2. How can you include comments in your program?

3. How do you designate fields as alphameric?

4. What rule or rules apply to the coding of calculation specifications, in regard to sequence of steps?

5. What operation codes are used to:

   a. Branch from point to point?

   b. Use an RPG II subroutine?

   c. Look up information in a table?

   d. Read a record from a disk file?

   e. Produce output during calculations?

   f. Locate logic errors during a test run?

6. In what order should RPG II statements be placed when ready for compilation?

7. Assume that halt indicator H1 was turned on during a calculation. When will the computer stop running?

3.3 – Self-Test Answers

1. Files are defined on the File Description and Extension Sheets.

   Records are described on Extension, Input and Output Sheets.

2. Comments statements may be included on any sheet except the control (H) statement at any point by placing an asterisk (*) in column 7 and comments following it.

   Brief comments may also be inserted in designated columns of the Extension and Calculation sheets.

3. Alphameric fields do not use an entry under "Decimal Positions".

4. There are three basic rules:

   a. Code detail, then totals, then subroutines.

   b. A "TAG" statement must be the statement to which a "GOTO" branches.

   c. After a subroutine has been used, the program "branches back" to the step that immediately follows the EXSR operation.

5. a. Branching – use GOTO and TAG

   b. Subroutines – use EXSR, BEGSR and ENDSR

   c. Tables – use LOKUP

   d. Read Records – use CHAIN, READ or FORCE

   e. Output – use EXCPT (and DEBUG during test runs)

   f. Logic Errors – use DEBUG (EXCPT is also useful.)

6. Statement order for compilation:

   H – control
   F – file description
   E – extension
   L – line counter
   I – input
   C – calculations
   O – output

7. After H1 turns ON, processing of that record is completed and output produced and then the computer halts; improper results may be prevented by conditioning appropriate calculation and output statements with a NOT H1 indicator.

3.4 – "RPG II Specifications Unique To System/32

As an experienced RPG II user, you will be pleased to note that very few specifications are unique to System/32 RPG II language. For convenience, they will be presented by specification sheet type. For specific coding rules, refer to the System/32 RPG II Manual, SC21-7595.

Control (H)

- Position 10, "Object Output" may be either blank or contain the entry D. In either case, the object program is written **permanently** into the object library. The D entry causes a halt if any warning errors are found during compilation. A blank halts only for terminating errors.

- Positions 19 and 20 serve a special purpose for System/32 users, a "Date Option".

  Position 19 may contain either M,D, or Y, or it may be left blank. Blank or M signifies the use of the month/day/year arrangement. A letter D indicates the arrangement day/month/year, while the letter Y is year/month/day.

  This arrangement should be the same as the one used for the system date.

  Position 20 is used to specify the particular character to be used as a separator when UDATE is part of the output and edit code Y has been specified. Any one of 256 characters, including blank, may be used as a separator rather than slashes. If both positions 19 and 20 are blank (and position 21 is also blank), the date will be edited using slashes. An ampersand (&) forces a blank separator.

- Position 21, "Inverted Print" is brought up here because it may affect the expected results from coding entered in columns 19 and 20. The general rule is, "if position 21 is blank, the entries in position 19 and 20 determined the format of UDATE when edited using edit code Y." Since an I, a J or a D can be specified for position 21, you must check all three positions if you desire something other than mm/dd/yy as output when editing UDATE. Keep in mind that entries in position 21 also affect the editing of numeric fields. There are many combinations of entries that are permitted (4 in position 19, 256 in position 20 and 4 in position 21), so consider them carefully before making entries. Also, examine all edited output during test runs to assure yourself that these specifications have been made properly.

File Description (F)

- Positions 20-23, "Block Length" and positions 24-27, "Record Length" are dependent upon the device used to read in or produce file records. The system has a keyboard/display screen combination that may be specified in three different ways as the file device in position 40-46.

Let's consider these one at a time.

- The input device named KEYBORD may be used to enter data as a "demand" or "primary input" file device. Record length is 40 characters or less. Block length is not used.

  Using KEYBORD actually ties both the keyboard and the display screen together as one device. Operator entries are displayed as they are keyed.

- The display screen output device named CRT may be used to display up to 6 lines of 40 characters each. Either normal or exception output may appear on the screen. Record length is 1-40 characters while block length is not used.

  A line of data moves onto the bottom of the screen and moves all other lines upward. The top line is lost from the display.

  Output operations of spacing and skipping may be specified for the CRT, but there are restrictions. "Skip Before" of 01 is acceptable and causes the screen to be erased. "Skip After" is not permitted. Editing may be used with fields in a CRT file.

- The device name CONSOLE is used to designate a file as an Interactive Data Entry (IDE) file. Section 3.10 provides specific information as to the use of and restrictions related to the use of the console file.

  Using CONSOLE actually ties both the keyboard and the display screen together as one device. Operator entries are displayed as they are keyed.

  Record length may be from 4-160 characters. Block length must be at least two more than the record length.

Input (I)

- Positions 19-20, "Record Identifying Indicators", positions 21-41, "Record Identification Codes", and positions 44-51, "Field Location" have restrictions when console file is being described. See Section 3.10 for details.

Calculation (C)

- External Indicators U1-U8 can be resulting indicators

- The SET and KEY operations (positions 28-32) are not new to users of System/3 Model 6 computers. However, you should be aware of new functions they can perform. Use of either requires definition of a KEYBORD file.

  The SET operation has two new functions: clear the input buffer area when using an Interactive Data Entry Console file, and display a user library message.

To clear the input buffer area, code as shown:

Positions	Entry
28-32	SET
33-42	the filename of the interactive data entry file
43-48	ERASE

This operation clears all records buffered following the one being processed. It resets to the record being processed.

To display a user message that is a user message member, code SETnn in positions 28-32. The entry nn must be the two-digit message identification code number that is needed to retrieve the desired user message.

The KEY operation may include an entry in positions 31 and 32. If used, this entry identifies the message identification code needed to retrieve a user message that will prompt the operator to perform a KEY operation. For example, KEY 16 will cause user message 16 to be displayed as a prompt. Factor 2 is not required for the KEY operation.

The SET and KEY functions described here may be used in combination at times. For specific coding rules and examples, you should refer to the System/32 RPG II Reference Manual, form number SC21-7595.

Both SET and KEY may include an entry in Factor 1 to identify data for the display screen. That entry may be a constant, a numeric literal, a field, a table element or an array element. Factor 1 takes precedence over a message identification code in positions 31 and 32.

Output(O)

The entries for positions 19-20 (Skip Before) and positions 21-22 (Skip After) are limited to a line number from 01-84 for printer files.

The skipping entries for display screen (CRT) output files are limited to 01 for Skip Before and must be left blank for Skip After columns. The Skip Before entry for CRT files performs a special function – clearing all data from the display screen.

****

The unique entries identified in Section 3.4 provide special facilities to System/32 RPG II users. In addition to the entries discussed here, read Section 3.10, "Interactive Data Entry" as it includes information regarding RPG II coding for Console files.

3.5 System/32 Self-Test

Use scratch paper to record your answers to these self-test questions about unique, System/32 RPG II coding entries. Do not refer to this Section as you take the test. When finished, check your responses against those given on the next page.

1. A "date option" entry may be made on the Control Specification Sheet. What three arrangements are acceptable to the system? What other entry on the Control specification may affect the date option entries?

2. What device name is used to identify:

   a. An input file entered from the keyboard?

   b. An output file shown on the display screen?

3. What is the purpose of the SET operation when ERASE is entered under the Result Field (43-48)?

4. What is the largest print line number that may be entered under "Skip"?

3.6 – Self-Test Answers

1. The data may be arranged either:

   a. mm dd yy

   b. dd mm yy

   c. yy mm dd

   Another entry that may affect the date option is found in position 21, "Inverted Print".

2. a. KEYBORD for an input file

   b. CRT for an output file

NOTE: The device name CONSOLE identifies the file as an interactive data entry file in which case **both** the keyboard and display screen are used as input.

3. SET with ERASE resets the CONSOLE file buffer. It clears all records which the operator may have buffered, up to the one being processed.

4. The largest print line for skipping is 84.

3.7 – System/3 Disk RPG II Specifications not available to users of System/32 RPG II.

The specifications included in this section reflect differences in systems hardware between the System/3 and the System/32. Because of this fact, the first topic is an introduction to the System/32.

System/32 Hardware

The System/32 is about the size of an office desk and includes: a processing unit, a keyboard, a cathode-ray tube display screen, print capability and ic magnetic disk storage.

Processing Unit

This unit contains 16,384; 24,576 or 32,768 bytes of main storage as well as housing the arithmetic/logic unit and the control section. The processing unit operates with input/output overlap as jobs are being run. This unit is located in the area under the display screen.

Keyboard

The keyboard contains standard typewriter keys for entering alphameric data, a cluster of ten numeric keys along with an "ENTER +" and an "ENTER -" key for convenience in keying numeric data, and a set of 15 "function" keys to control the entry of data, positioning the cursor (underscore) on the display screen, restoring paper position and so on. In addition to these items, the top row of 12 typewriter section keys can be defined as Command keys to control program functions. To the right of the keyboard is an operator panel. It contains the Power On/Off switch, the Load (IPL), Start and Stop keys, and four indicator lights (Keyboard Ready, Processor Check, Thermal Check and Power Check).

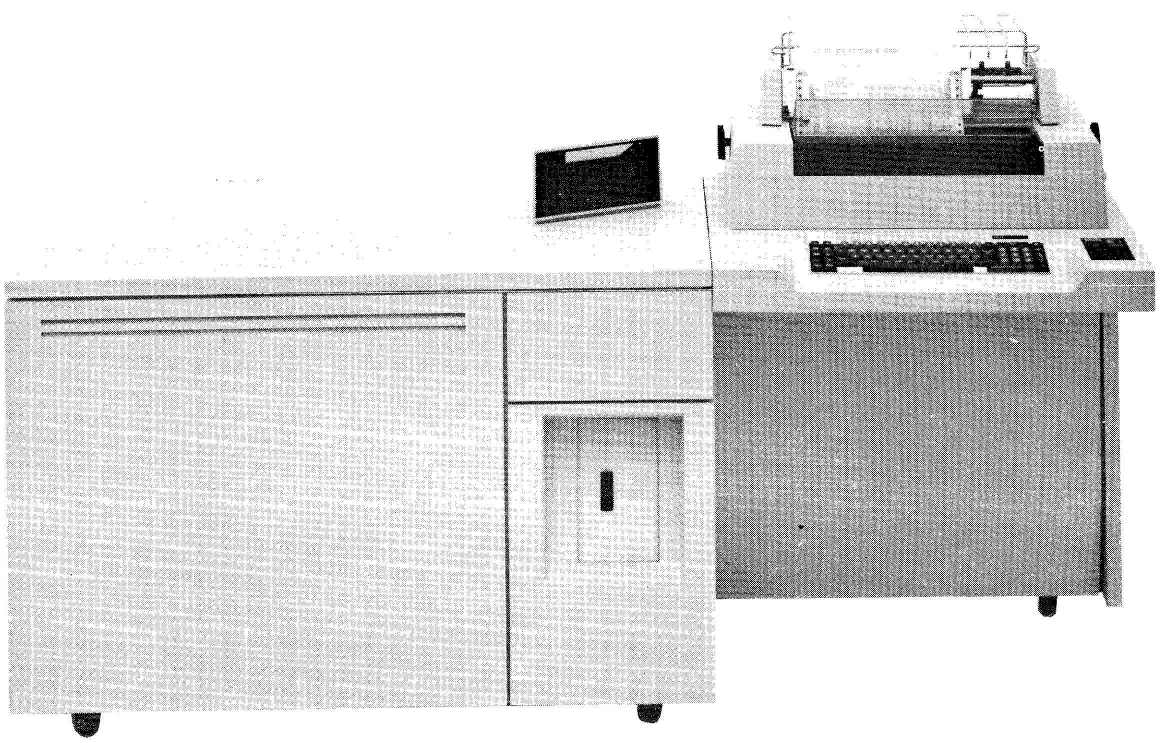
Display Screen

This unit is used to display input/output records and present messages to the operator. Up to six lines of 40 characters may be displayed at one time. A log of system history may be displayed here for operator convenience. The history may be "rolled" to find out what action was taken at an earlier point in time. A cursor (underscore) is used to indicate the display position into which data will be keyed. The displayable characters include all of those on the keyboard plus a reverse slash.

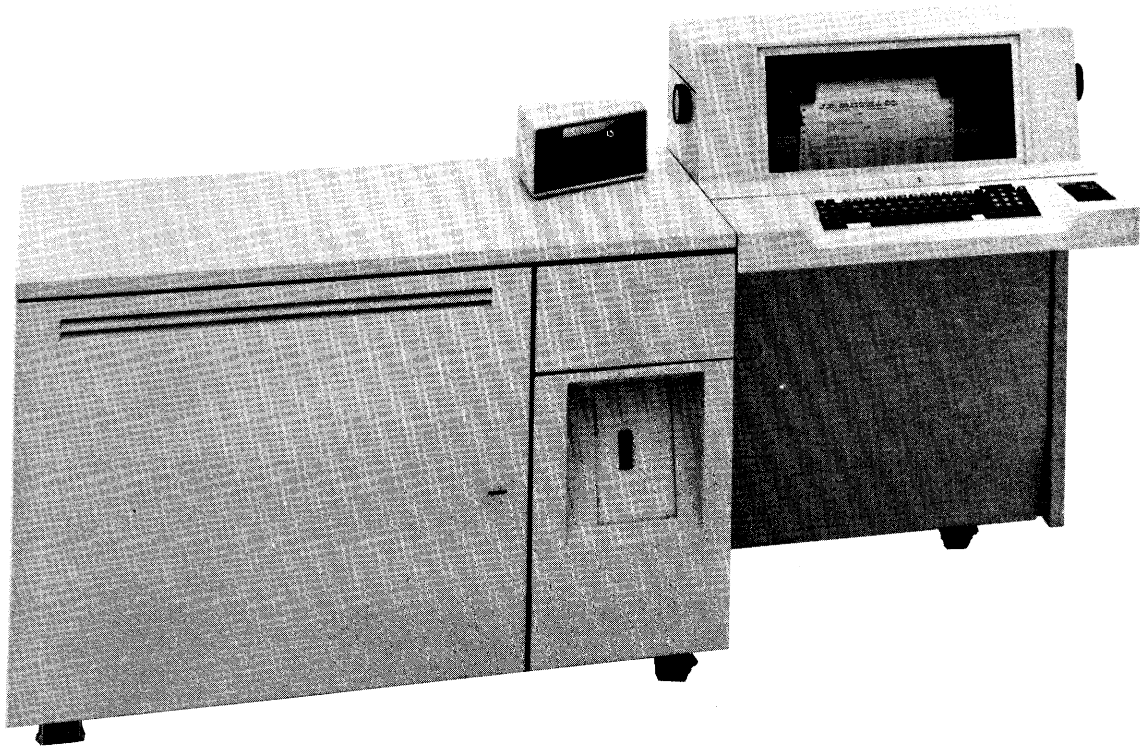
Print Capability

A variety of printing speeds ranging from 40 characters per second to 155 lines per minute are available on different models of the System/32. Both the serial printer and the line printer space vertically at six (6) lines per inch and can print up to 132 characters per line spaced ten to the inch. Multiple copies may be printed at one time by using multi-part forms. Thc line printer may have either a 48-character set or 64-character set on its embossed print-belt. Print-belts are interchangeable.

3740 Data Entry System



System/32 With Serial Print Capability



System/32 With Line Print Capability

Magnetic Disk Storage

A disk is located in the area under the display screen. It can hold system control programs, user applications programs and data files. The capacity may be either 5,038,080 bytes or 9,154,560 bytes. Programs and/or data will be moved from the disk to the processing unit prior to use.

System/32 provides for removable magnetic disk storage in the form of a "diskette". A diskette is approximately eight inches square and less than one-sixteenth inch thick. It holds a total of 246,272 bytes of information that may be read or written by the System/32. Diskettes are placed into a slot at the front of the processing unit near the floor. When the diskette has been loaded and the door is closed, the operator can cause the system to read data from the diskette to the disk, or write data from the disk onto a diskette. The programmer will code only the disk as a device (DISK). System/32 utility programs transfer data to and from the diskettes. Multiple diskettes may provide input data or store output data for a given job. Diskettes may be created on the IBM 3741 Data Entry Station for input to the System/32.

For additional information about System/32 you may wish to read manual GC21-7582, "Introduction to System/32."

System/32 RPG II Specifications Not Available

A number of RPG II entries refer to specific system hardware. The IBM System/3 may make use of punched card or magnetic tape I/O devices while System/32 does not have these facilities.

Here is a list of punched card and magnetic tape-related entries not available for your use.

SPECIFICATION	COLUMNS	
H	44	Punch leading zeros
F	15	Combined (card) files
F	19	Variable length tape records
F	40-46	Card device: MFCU1, MFCU2, DATA96
F	40-46	Tape Device: Tape
F	54-59	ASCII tape file is referenced
I	19-20	Spread card records
I	42	Card stacker pocket
O	16	Card stacker pocket
O	32-37	Print data on cards
O	40	Print data on cards

In addition to the items related to punched cards and magnetic tape, consider the following points:

1. The operation code DSPLY is not used. In its place, the System/32 uses the KEY operation.

2. The operation SET does not perform the functions of SPACE, SKIP, EJECT or tab settings as it did on the System/3 Model 6.

3. No ledger card support is provided.

4. There is no Sterling conversion entry.

Since there are no punched card I/0 devices on the System /32, you may have thought of the problem involved in converting coded specifications into source programs. A utility program named the Source Entry Utility (SEU) is used for that purpose. The operator calls SEU and selects a pre-determined format for each sheet of RPG II code. As entries are keyed, they are displayed on the screen for visual verification. As each statement is entered, SEU "builds" and stores the source program. After keying the program, it may be used to modify existing source programs. The operating procedure for creating, modifying and compiling an RPG II program is found in Appendix 3B of this book.

For specific information refer to the System/32 Source Entry Utility Reference SC21-7605.

3.8 Self-Test

1. List four devices not available on System/32 for which System/3 RPG II coding is acceptable.
2. What equivalent code may be used to perform the DSPLY function?
3. Describe the System/32 hardware briefly in your own words.
4. How is an RPG II source program created in System/32?

3.9 Self-Test Answers

1. System/3 devices not available to System/32 users include:

   a. 5424 MFCU

   b. 5496 Data Recorder

   c. 3410/3411 Tape Drives

   d. 2501 80-column card reader

   e. 1442 80-column card read/punch

   f. 5471 Console typewriter

   g. 5444 Disk Drive

   h. 5445 Disk Drive

   i. Ledger Card Device

   j. Dual Printer Carriage

2. The KEY operation provides for the function that DSPLY serves on System/3.

3. The System/32 contains 16K, 24K or 32K storage, a serial or line printer, a display screen, a fixed disk of 5 or 9.1 million bytes of storage, a keyboard and a diskette drive for input/output.

4. The source entry utility program provides a facility for creating and storing System/32 RPG II source program.

3.10 The Console File For Interactive Data Entry (IDE)

This function of the System/32 RPG II language allows the operator to enter data to an executing RPG II program via the keyboard. First, the operator calls the desired program using appropriate OCL statements such as:

```
// LOAD JOBA
// RUN
```

When the program has been loaded, the display screen "prompts" the operator to enter data from the keyboard. The prompt looks similar to this:

```
15\

COST      N04.2      1
////
```

The characters at the upper left represent the record identification code (or codes). In this example, position 1 has a code 1 while position two (2) has a Code 5. The reverse slash indicates that a field of data will be keyed and displayed here for reference.

Below that, a line shows the field name (COST), whether it is numeric (N) or alphabetic (A), the size of the field (04.2) and at the far right, a number to indicate the record type being prompted (1). For this example, the operator is to key data into a field of the first record type. The series of slashes on the lower line will be replaced by data as it is keyed on a character-by-character basis.

When the field (COST) has been entered, the prompt will change in this manner. First, the data keyed for COST will appear on the top line to the right of the reverse slash, and another reverse slash will appear to its right if another field is to be keyed.

Next, a new field name and description will appear below. Also, the number of slashes that represent the new field size will appear on the lowest line. Again, the operator keys in data on a character-by-character basis for this field and then enters it. Keying continues in this manner until a record has been entered. At that time, output is produced as prescribed in the program while the operator keys in data for the next record. When an operator is entering data in a file that has two or more record types, it may be necessary to switch from one type to the other. To do this, press CMD and a number that represents the record type. That number will be the same as the record identifying indicator number you specify on the Input sheet. For example, if you specified indicators Ø2 and Ø7, the operator must press CMD and a 2 or CMD and a 7 to select the proper formats. This procedure continues until all data records have been entered by the operator. To complete the job, the operator presses the CMD key followed by the slash key. This turns on the last record indicator (LR) and the RPG II program terminates when all specified steps have taken place.

To review, when the operator runs an IDE program, the steps are:

1. Load the desired program,
2. Key in data as fields are prompted,
3. Select new record type formats if necessary, and
4. Terminate the job.

Now that you've seen how the operator uses an IDE file, let's consider the RPG II coding that needs to be specified.

File Description Sheet

Most entries here are similar to those used for other input files: File Name, File Type, End-Of-File, Sequence, and File Format for example.

Here are the three key differences:

1) Device name (40-46) must be CONSOLE. Only 1 IDE file may be specified for a job. When CONSOLE is coded, both the keyboard and the display screen are treated as one device.

2) Record length (24-27) is calculated from a formula. The minimum record length is 4, while the maximum is 160. Here's how it is calculated:

RL = Highest input field end position + number of specified input fields + 1.

3) Block length depends on the record length for its size. It must be at least two more than the record length. Large block lengths accomodate high speed operators by providing more buffer space.

Input Sheet

Again, entries here are similar to those used in other RPG II programs. Considerations when using IDE become simply a set of limitations as listed below.

The entries for *record identifying indicator* (19-20) define the number of the command key to be used when an operator selects a record type to be entered. The limitation is Ø1-1Ø. For *record identification codes* (21-34), RPG II requires that for IDE files it be in position 1 of the record or in both positions 1 and 2. All record types of the console file must be identified and there must be at least one field for every record type. Your codes should be:

Position	Entry	Note
24	1	Required
25	Blank	
26	C	

27		Use any letter, number or special Character
31	2	If used, must be a 2
32	Blank	
33	C	
34		Any character

When defining fields (44-70) remember this rule.

Each succeeding field that is specified must be contiguous (no unused spaces between fields) to the previously specified one, or it must be a "sub field" in effect of another.

EXAMPLE 1: Contiguously specified fields:

```
IUSUAL  AA  01  1  CX
I                        2       5 1 AMT
I                        6       9 4 RATE
I                      1 0     2 2   NOTE
```

EXAMPLE 2: Contiguously specified fields plus "Sub fields".

```
ISUB  XY  01  1  CS
I                        2       5 1  AMT
I                        6       9 4  RATE
I                        8       9 2  RSF
I                      1 0     2 2    NOTE
I                      1 8     1 9    SCD
```

Examine these fields. Are they acceptable?

```
ITHRE  ZZ  01  1  C-
I                        2       8  A
I                        9     1 2 3 B
I                        6       7  C
I                      1 1     1 1 1 D
```

Yes, because the sub-fields C and D are parts of previously specified fields A and B.

What is the error in this coding?

```
IEROR  NS  01  1  CM
I                        2       7 2 P P L
I                        8     1 2 3 X E R
I                      1 3     1 5   C D
I                      1 4     1 8   G R P 6
```

Fields PPL, XER and CD are specified contiguously, so they are fine. Field GRP6 is in error. First, it is not contiguous. Second, it is not a sub-field of any others previously specified.

Evidently the input record is not correctly defined and needs to be changed so that it can be specified as an IDE file record.

Three additional restrictions regarding the use of IDE files must be remembered when coding Input specifications.

1. You are not allowed to specify AND (14-16) for IDE fields,

2. No entries may be made for field record relation (63-64),

3. Input fields that are alphameric cannot exceed 40 characters.

Summary

In summary, the Interactive Data Entry function allows data to be keyed from the system keyboard, displayed for reference on the CRT Display Screen, and accepted for processing by an executing RPG II program. One IDE file may be defined in a program. Records in this file may be of various types and be from 4-160 characters long. Special coding rules apply to entries on the File Description Sheet (Block Length, Record Length and Device) and to entries on the Input Sheet (Record Identifying Indicators, Record Identification Codes and Field Location). Every program that uses an IDE file provides "prompts" on the CRT Display Screen for operator convenience and reference. To terminate the entry of data in this kind of program, the operator presses CMD and the slash key. If CONSOLE is named as a Device, RPG II automatically generates an Interactive Data Entry program.

For additional information and examples, refer to the System/32 RPG II Reference Manual, SC21-7595.

3.11 Self-Test

Try to answer these questions without reference to this section or the RPG II reference manual. Answers are provided on the next page so that you may check your own responses.

1. How does the programmer provide display screen "prompts" to the operator for keying CONSOLE file data?

2. What is the shortest record length allowed for CONSOLE files? The longest?

3. A CONSOLE file was described as shown below. What is the minimum record length you could specify for this file?

```
IABC  NS  Ø1  1  CT
I                       2     5 1 NUM
I                       6    1Ø 2 VALUE
I                      11    25   EXPL
```

4. What must the operator do to terminate a job after entering all IDE file data provided?

5. How many record identifying indicators may be specified for an IDE file? Which ones are acceptable?

6. Where must record identifying codes be placed in IDE file records?

3.12 Self-Test Answers

1. By simply specifying record fields, the generated RPG II program creates the prompts.

2. Shortest record length is 4 characters.
   Longest is 160 characters.

3. Minimum is 29 characters in this example.
   RL=Highest field end position plus number of fields plus 1

   =25+ 3 + 1
   =29

4. Press CMD and a slash.

5. Up to 10 record identifying indicators may be specified.
   Acceptable entries are Ø1-1Ø only.

6. Either in position 1, or in positions 1 and 2.

****

At your convenience, read the section on the CONSOLE file for interactive data entry in the System/32 RPG II Manual SC21-7595. Also, take time to examine Appendix 3A which describes the Auto Report feature and Appendix 3B which describes System/32 Operations used to compile an RPG II program.

Appendix 3A-The RPG II Auto Report Feature

The Auto Report feature accepts special RPG II-like specifications along with normal RPG II specifications in order to generate a complete RPG II source program.

The special RPG II-like specifications can provide for:

1. A report page title that is automatically centered on each page and may include the date and page number,
2. Columns of data with corresponding column headings and totals as desired, and
3. The copying of frequently used specifications already cataloged into the library.

A report title can be coded in this manner using the Output Specification Sheet. First, code the output file name in columns 7-14, the letter H in column 15, and the entry *AUTO in columns 32-36 on the same line. Next, on a new specification line, code the title as a constant in columns 45-70. Use additional constants if the report title is a long one.

EXAMPLE: To specify a report title of MASTER INVENTORY LIST, the specifications would be:

```
OREPORT    H      *AUTO
O                              'MASTER INVENTORY LIST'
```

Notice that no entries were coded for Spacing, Skipping or Output Indicators. Notice also that no entry is made for End Position in Output Record for the title and that no entries were required for including the system date or page numbers.

These two special RPG II-like statements will produce generated statements for a report title such that:

1. The system date will be included on the left,
2. The report title will be centered,
3. The constant, PAGE, will be printed at the right of the title followed by the page number,
4. The title will print on all pages under control of indicators 1P and OA; printing will start on line six and double spacing will follow.

Now let's examine coding that might be used to print out five columns of data. The report is expected to look something like this:

```
      6/18/75    MASTER INVENTORY LIST     Page 1

ITEM           ITEM          UNIT                           TOTAL
NUMBER         DESC          PRICE           QUANTITY       VALUE

  106        FLAVORLIFT        1.27              6             7.62
  129        ERADICK            .45             29            13.05
  466        BIGG CAT        175.64             12         2,107.68
                                                            2,128.35*
```

To specify these entries, assume that the fields are named ITEMNO, DESC, UP, QTY and VAL on an input specification sheet. Here is how to code the the entries using Auto Report.

First, describe the record as a detail output line (let's assume indicator 01 was used and we desire single spacing).

```
0  D  1  01  *AUTO
```

Next, describe each field to be printed, *and* include the column heading desired as a constant, *and* indicate that the one field (VAL) is to be accumulated. Here is how they are specified:

```
0        ITEMNO                 'ITEM'
0                       C       'NUMBER'
0        DESC                   'ITEM'
0                       C       'DESC'
0        UP                     'UNIT '
0                       C       'PRICE'
0        QTY                    ' '
0                       C       'QUANTITY'
0        VAL            A       'TOTAL'
0                       C       'VALUE'
```

Each output field is described in order of its appearance on the report, from left to right. If a column heading is to be included, it is entered on the same line as a constant. In this example, column headings use two lines of coding. To specify the second header for the field, enter a C (Continuation) in column 39 ("Blank after" column) of the next line and specify the header as another constant. The first three fields in the example are coded in this manner.

On the report sample, the quantity field is to be designated by a single heading constant, QUANTITY, on the *second* line of column headings. To specify this, enter the field name QTY and a constant of one blank on the first line, and then specify C in column 39 and the desired constant on the second line. Now examine the entries for the value field. An A is coded in column 39 on the same line that contains the field name. This code directs Auto Report to accumulate a total for the field name VAL. Also, specified for this field are the two columnar headings. Note that no field end positions or edit codes codes are specified. Auto Report will arrange for both conditions automatically during program generation.

Take a look at the heading constants for the field named UP. The first is UNIT and the second is PRICE. In order to print the report with the word UNIT aligned on the left, one space is included on the right as part of the constant. As you can guess, to align it on the right with the E in PRICE a space could have been included as the first character on the left.

This example shows some of the capabilities of the Auto Report feature of the RPG II language. In this case the programmer would code:

1. A control card specification,
2. Two file description specifications
3. Six input specifications using normal RPG II code.

You would also code 13 special Auto Report output specifications. As a result of your coding, the generated program will include additional RPG II specifications that provide for:

1. Page overflow
2. Report date
3. Page numbering
4. Report format
5. Edit codes
6. Final total
7. Calculations of the final total.

For this example, Auto Report generated an RPG II source program of 42 statements from the original 22 statements.

You have seen how Auto Report can provide for report page titles and report formats. Now let's examine another capability: the copying of frequently used specifications that are already cataloged in the library. We will assume that the file description specification and the input specifications are going to be used by our department in many applications. They have been cataloged as a library member named DATUM so that every programmer may have access to them.

When you are ready to use this set of cataloged specifications, you code a /COPY entry where it is appropriate. In our example it could be used as shown here.

```
H                                                JOBXYZ
F/COPY F1, DATUM
FREPORT  O         132          PRINTER
OREPORT  H            *AUTO

Etc.
```

The correct coding is as follows:

COLUMNS	ENTRY
7-11	/COPY
12	BLANK
13-15	F1,
16-23	Library member name as cataloged

The /COPY statement may be placed wherever convenient, provided that it is after the H specification. In our example, we placed it with the file description specifications (F in position 6).

In other instances, you may find it convenient to place a /COPY statement with calculations, e.g., when the library member does a particular sequence of calculations that you will use in your program.

Examine the next set of Auto Report specifications to see how a number of /COPY entries might be used.

```
H   P                                  EXMPL
F/COPY F1,INFRE
F/COPY F1,OUTMANY
C 01             CUMR   MULT 6.28      ACC 51H
C/COPY F1,CALSTAT
CLR              ACC    COMP 1259.6
CLR 2 7                 Z-ADD1259.6    ACC
```

Three /COPY entries were made to illustrate placement only. This example shows one library member (INFRE) that probably included one File Description and a number of Input Specifications. Another library member (OUTMANY) that probably includes one File Description and a number of Output Specifications. A third library member (CALSTAT) evidently includes a number of Calculation Specifications.

When Auto Report generates an RPG II source program from such a sequence, each specification is placed with the proper form type before compilation takes place.

Now, let's review the three points made about special, RPG II-like specifications.

1. A report page title may be specified by including *AUTO in positions 32-36 of an "H" type of line. This title will be centered and the date and page number included with it.

2. A report page format may be specified by including *AUTO in positions 32-36 for a "D" type of line. Fields, and associated column headings are specified following this entry. Numeric fields will be edited and may be accumulated.

3. Frequently used sets of RPG II specifications may be cataloged and stored as library members on the system. One or more members may be copied from the library to be included in programs wherever and whenever convenient.

This brief examination of the Auto Report feature does not do it justice in that many additional uses exist. For full details, you should refer to the System/32 RPG II manual, SC21-7595.

In addition to the information presented, you should know about the Auto Report Option Specification because it allows you to assign a unique program name to the generated RPG II source program, thereby having the ability to access it and make modifications (change edit codes, re-position fields, etc.) prior to final compilation. If used, this statement is first, then the H specification and so on. Here is its makeup:

POSITION	ENTRY	EXPLANATION
6	U	Identifies specification as Auto Report Option
7	C	Catalogs generated source program
8-10	F1,	Library location (fixed disk)
11-18	Library Name	The name you assign for accessibility
27	Blank	Include the date and page numbers on the first *AUTO page heading line
27	N	Suppress the printing of date and page numbers from the first *AUTO heading line
28	Blank	Include asterisks behind totals
28	N	Suppress asterisk printing for totals
75-80	Program Name	

Note: If you do include the Auto Report Option specification, be sure that the program name also appears in the H (control) specification. If no program name is included in a control (H) statement, the generated program is named RPGOBJ and will be replaced by the next program that is also unnamed.

Summary

The Auto Report feature of RPG II is an aid to the programmer in that it generates additional RPG II statements and/or copies statements already present in the system library.

This feature simplifies coding in that:

1. Fewer statements are coded,

2. Report format planning is reduced,

3. Repetitive coding may be copied,

4. Totals are accumulated with a minimum of code.

As a result, fewer errors reduce program debugging time and shorten the span from plan to run.

The RPG II source program generated by Auto Report becomes accessible for program additions and/or modifications prior to final compilation by using the Auto Report Option specification.

When an RPG II program includes one or more Auto Report entries, the operator keys AUTO followed by a space and the name of the program. If no terminating errors are diagnosed, this program will be ready to run using test data. For more information on operating the system, refer to Appendix 3B.

Appendix 3B-Operating The System/32

The information in this appendix is intended for the programmer who will be entering source programs, compiling them and testing them.

The first page should be adequate for programmers who have keyed in a source program in the past.

The remaining pages provide a more detailed step-by-step procedure for the first-time user.

System/32-Operating Procedures

Using SEU, AUTO or RPG

To ready the system:

- Turn power ON
- Ready the paper in the printer
- Press LOAD
- Enter the system date

Using SEU to enter a source program:

- Key in SEU, a space, the program name, a comma, the letter *A* if Auto Report statements are part of the program *or* the letter R if only RPG II coding is used, and ENTER.
- Use CMD and 6 to establish print status; CMD and 8 for syntax status
- Select desired format: CMD, 3, letter code, and Enter
- Key entries, use Enter and REC ADV
- Select next format as above, and key entries
- After last source statement, press CMD and 7
- Choose from menu, and Enter

Using SEU to update a source program already entered:

- Key in SEU, a space, the program name, A or R and Enter
- CMD, 6 for print status; CMD and 8 for syntax status
- Key in desired statement number, and Enter

- FIELD ADVANCE correct field entries in statement
- Key change (s) and Enter for each
- REC ADV at end of each corrected statement
- End update activity with CMD and 7
- Choose from menu, and Enter

Compiling a source program already entered:

- Key in either AUTO or RPG, a space, the program name, and Enter

Note: RPG defaults to 20 blocks for $SOURCE and $WORK parameters unless you key in other values. The RPG statement is:

RPG source name [,mmm] [,nnn] where mmm is $SOURCE and nnn is $WORK parameter.

- Examine printout diagnostic messages
- Make appropriate changes and recompile
- Run test data: use Enter- key for negative fields

Power off:

- Remove printout
- Turn power switch OFF

System/32 Operating Procedures Detail

- Readying the system
- Using SEU to enter an RPG II program
- Using Auto Report to generate and compile a program entered under SEU
- Testing a compiled program
- Turning the system Off.

Readying the System

- Turn power ON; switch is at the far right of the keyboard
- Make certain the paper is in the printer and it is ready for use

- When the STOP key/light turns ON, press the LOAD key

  In about 30-40 seconds, a message will display information about the system status including the system date.

- If not yet done, enter today's date by keying

  DATE mmddyy (mm is month, dd is day and yy is year)

- Press the ENTER key on the right side of the keyboard

NOTE: If at any time an unacceptable error is made while you are entering information, the display screen will flash repeatedly. Press the Error Reset key on the left side of the keyboard and examine the display. If the cursor (underscore) is not visible, you should press the Field Backspace key (marked Field BKSP) on the top row on the right side to re-position the cursor before keying a correction.

Using SEU (Source Entry Utility) to enter an RPG II program into the source library.

- When the display screen is ready (READY is shown on the lower right), key in the letters SEU, one space, then the name of your program (on the coding sheet in columns 75-80), a comma, and either the letter A (if Auto Report entries are part of your program) or the letter R, and then press the ENTER key. Use the SEU template for command key function identification.

- Shortly, the screen will show a display similar to this:

```
001  0  A096  0001.00  S



ENTER/UPDATE STATEMENT NUMBER: 0001.00
```

- Examine the display. The top line is called the status line and shows, from left to right:

– A number - indicates the position ready to accept keyed data

– A letter or number - indicates the "format" SEU assumes for you (you will select other formats as you enter data)

– A letter followed by some number - indicates the type of field to be keyed in, and its length

– A six-digit number with 2 decimal places - indicates the statement number to be entered

– One, two or three letters (or none at all) - indicates the mode (s) under which SEU will treat your keyed entries:

A - indicates that designated fields will be duplicated or skipped automatically as keying takes place

S - indicates that SEU will provide syntax checking of the entries you key prior to compilation

P - indicates that each statement will be printed out after being keyed

– On the middle line, left side, you will find the cursor. It indicates the position on the screen into which keyed date will start to display.

– On the bottom line you will see the message

ENTER/UPDATE STATEMENT NUMBER: 0001.00

This number is used by SEU to keep track of the entries as you make them in the Enter portion of the Enter/Update mode.

- If the letter P does not show in your display, press CMD and then the 6 key in the top row on the keyboard. The letter P should now appear. Also, if the letter S does not show, press CMD again and then press the 8 key on the top row. The letter S should now appear, and you are ready to enter your source program.

- To key in program specifications, you select a format which treats your information as a program card would be handled on a keypunch, and proceed to enter the specifications as coded. The pattern for entering source program information is:

  Press the command key (CMD) on the top row, then press the 3 key on the top row (this alerts SEU to find the desired format and makes it available to you), then press the letter of the format you plan to use, and finally press the Enter key to complete the selection.

  Here is the list of formats from which you may select.

Code	*Format*
U	Auto Report Option
H	Header control statement
F	File Description
K	Auto Report copy (/COPY) statement
E	Extension
L	Line Counter
I	Input . . . to describe input records, but not their fields

J Input . . . to describe input fields only

C Calculation

O Output . . . to describe output records, but not their fields

P Output . . . to describe output fields and constants only

Ø (zero) an "all purpose", 96-position format

NOTE: Additional information regarding formats and the use of SEU may be found in the Source Entry Utility Program Reference Manual SC21-7605.

- After selecting the desired format follow this pattern:

  1. Press the ENTER key to skip a "field" on your RPG sheet where no entry was coded.

  2. Key coded entries on a field-by-field basis, and press ENTER for each.

  3. After making the last entry for a line of code, press the REC ADV (record advance) key on the top row at the right sides. This causes the keyed entry to be stored by SEU for later compilation, and advances the statement number count to the next whole number. If CMD and 6 were used, the entry is also printed.

Entering your program — refer to your program and enter it now

- Select the H format (or the U format if an Auto Report Option is used in the program): Press CMD, 3 on the top row, letter H (or U) and ENTER.

- Press the enter key until the position counter (upper left on the CRT) is at the place where your entry should go, then key in the entry you coded on the RPG sheet, and press the enter key. SEU automatically inserts the letter H in position 6, so you only key in what you code. SEU also stores the program name you keyed earlier (see item 1 on a prior page) when you called for SEU. The program name should show in positions 75-80 of the screen format for the header statement.

- Press REC ADV (upper right) to complete the header specification. If the letter P is shown at the upper right on the screen, the entire header statement as you keyed it will be printed along with its number, and SEU is ready to receive your next line of code.

- Select the next format: CMD, 3 on the top row, letter F if you want File Description, and Enter. Key in your coded entries field-by-field, and press REC ADV at the end of the line. Continue in this manner for your entire program.

NOTE: Some statements may not be accepted as keyed (syntax checking error), yet are keyed as you coded them. When the screen flashes to indicate this condition, examine the keyed entry; if it is keyed correctly, press Error Reset, CMD and the plus +key on the top row. Your program will now include the entry.

Correcting Keying Errors

If you make an error while keying a field (you notice it on the screen), simply press the FIELD BKSP key on the upper right on the keyboard and re-key the correct entry, and then press the enter key.

If you make an error and realize it after REC ADV has been pressed, you will have to update this line of code prior to compilation using the Update portion of the Enter/Update mode of operation. This procedure is explained later on, so just continue with your keying on your next RPG II statement.

Changing Formats

- Press the command Key (CMD), the 3 key on the top row, the letter code for the format you wish to use, and the Enter Key in that order.

- Enter your File Descriptions at this time. The pattern:

  CMD, 3, letter F, enter

  Enter, to locate the position for filename
  Key the entry and press ENTER
  Continue with the next field; if no entry is coded, just press Enter

  After entering the last coded entry for the line, press REC ADV.

- When entering Input specifications, select format I for record descriptions (columns 7-42) or J for field descriptions (columns 43-74). SEU inserts an I in position 6 for each of these items.

- Use format C for calculations.

- For Output code, select format 0 (letter 0) for record descriptions (7-37) or P for fields and constants (23-74). SEU inserts letter 0 in position 6 for each.

Completing the Program

- Press CMD

- Press the 7 key on the top row. This indicates to SEU that you are at the end of the job in regard to entering a source program.

  This also causes a "menu" to be displayed. The menu includes choices from which you make a selection. If this is the first program you have ever entered, it is suggested that you select 4. This choice provides for serial numbering of the coded statements as you have entered them, and it will print out a complete list for your reference. Press Enter to complete your selection.

- The next message asks you to indicate a position into which you wish to have the serial numbers inserted. Beware! If you choose a location that contains coded information, such as position 7 (filename, etc.) the serial numbers will overlay the coded entries and you must re-key every line of code before compiling can be done!

  SEU assumes that you want serial numbering to start in position 1 if you simply press the enter key in response to this message. This is a good way to handle it because the serial numbers will go into positions 1-4 which are not used for any RPG II source entries except page and line numbering.

  Press the enter key to complete this job and you will get a program listing for your reference.

Making Changes to a Program Already Entered

You will make use of the Update function of Enter/Update mode in order to make changes to stored specifications.

- Key in SEU, a space, the name of the program to be updated, a comma, and the letter A (for an Auto Report program) or the letter R.

- Press Enter; a display such as this will be shown:

```
                    0001.00    S

ENTER/UPDATE STATEMENT NUMBER: –
```

- If you wish to print changes as you key them and P is not shown on the screen, press CMD and 6 on the top row before making a statement change.

- Key in the number of the statement to be updated. If you key in the wrong number, press REC ADV which "puts it back" as it was, and lets you pick the one you really wanted.

- Press the Field Advance key (left side of keyboard) to duplicate each field that is to be retained as is, and then key in the correction(s). When the last correction for that line has been keyed and entered, press REC ADV.

- To insert a step that you may have overlooked when keying originally, for example, suppose you left out a calculation between statement 0016 and 0017, you key in a number from 16.01 to 16.99 such as 16.5 and then select the format you need (it may be the one that is displayed) and enter the missed specification.

- To add a new statement at the end of your program, key in a number that is just 1 higher than the number of the last statement in your program. Select the desired format, and enter the specification. Continue until all required statements have been added.

  If all that you have are additions to be placed beyond the last program statement, press CMD and 9 (Search End of Source) on the top row. This will locate the last source statement of your program. Key in a number just 1 higher, press CMD and 2 on the top row (Multiple Statements), and enter the first new statement. Continue entering statements until all have been added. This approach eliminates the need to know what your last statement number is before you can add a new one.

To go to the "end of job" for updates, press CMD, 7 on the top row, and take a choice from the menu. Press the enter key and then press enter if you want to start serial numbering in position 1.

Compiling Your Program

- Programs not using Auto Report:

  Call the RPG compiler by keying in the word RPG, followed by one space, and your program name.

  About 20-30 seconds is used to locate the compiler and your source program, after which time compilation will begin.

- Programs in which Auto Report statements are used:

  Call the Auto Report feature to generate and compile your program by keying in the word AUTO, one space, and the name of your program.

  In about 20-30 seconds generation, followed by compilation will begin.

- Examine the printout of your program for diagnostic messages. If there are no terminating errors (T messages) the compilation will continue to completion.

- If there are terminating errors, take option 3 by pressing 3 and Enter. Then follow the procedure for making changes shown in step 7 and try to compile again.

Testing a Program

- Load your program using appropriate OCL entries. Include FILE statements for any disk input/output files specified in the program. If no disk files were used, the OCL you would enter would look like this example in which the compiled program ABCD is to be tested.

  ```
  // LOAD  ABCD  (and then press Enter)
  // RUN         (and then press Enter)
  ```

- Enter data from the keyboard if you specified the CONSOLE as an input device: The "ENTER -" key (on the right side of the keyboard) is used to enter negative field values.

- Examine the test run output. If acceptable in all details, the program is ready for use. If unacceptable errors exist, you need to re-code, use SEU to update the program, re-compile it, and re-test it with the same test data again.

Power OFF

You may desire to (or need to) turn the system off after a run. To do this, simply turn the power switch to OFF. Allow about 20 seconds for the disk to stop spinning and the cooling fans to stop running.

GB 30-0001-0

Concepts of System/32 SCP, Utilities, RPGII for the Experienced User GB 30-0001-0 12-74 Printed in USA

IBM®

International Business Machines Corporatio

General Systems Division
5775D Glenridge Drive N.E.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

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