Pg 6

IBM

MSC 80-14

July 24, 1980

Distribution:

Regional Management and Staffs
Branch Office Management
Marketing Representatives
Systems Engineers

Robert M. Umbreit, Division Director
Product Programs

E. L. (Ed) Matthys, Manager
Rochester Market Support Center

International Business Machines
Corporation
Market Support Center
Department 938
3605 Highway 52 North
Rochester, Minnesota 55901

IBM Internal Use Only

MSC Market Support Centers

Market Support Centers
GSD NEWSLETTER

IBM 5280 DISTRIBUTED DATA SYSTEM

- DE/RPG:
  - Operating Characteristics
  - Explanations
  - Techniques
- Utility Notes
- General Notes

CONTENTS

Page

DE/RPG--OPERATING CHARACTERISTICS

DE/RPG--EXPLANATIONS

DE/RPG TECHNIQUES

DE/RPG--OPERATING CHARACTERISTICS

The items in this section explain how certain functions operate in order to avoid incorrect coding based on misunderstanding of the function.

Compiler Workfiles

The compiler uses two workfiles, SYSUTOX1 and SYSUTOX2, where X is the partition where the compiler is executing. If these workfiles are not preallocated, the compiler will take as much space as it needs when it does the allocation. This is accomplished by opening the file to the maximum space left on the diskette, then releasing the excess on close. Should a manual EOJ be forced by the operator before a close has released the excess space, the compiler will terminate leaving the space allocated, and hence the disk will be full. SYSLABEL or SYSCLEAR must be used to delete the workfile and recover the space.

If a normal compiler termination occurs, and the compiler allocated the workfiles, it will delete them. However, the compiler start-up time can be reduced, and the disk full situation prevented, by preallocating the workfiles. 200 records of 128 bytes for each workfile is adequate for at least a program of 420 source statements and 28K of object code.

Object Size Grows on Recompilation

As programs are enhanced, the size of the object program may grow larger, to the point where it no longer fits into the previous space. The compiler will post a 3411 error at the time it writes to the object data set. Upon reset, it will prompt for a new name for the object data set. The easiest approach is to provide a new name and let the compiler finish. Then run SYSLABEL to delete the old version of the program, and then change the new name to the desired name.

Rerun Mode

Since the 5280 is an operator controlled system, it is possible for the operator to move randomly through the transaction file and make changes that could render the online field total values or calculation results inaccurate. The rerun function is used after normal data entry to reprocess the transaction file in sequential fashion with no operator keying. All edits and calculations are reperformed, and all results corrected to the proper value. Transaction records are automatically updated with the corrected values. Errors are posted to the operator, who can accept or mark the field in error and continue. The errors can be corrected in update mode, then rerun can be used again to recalculate values.

Rerun mode can also be used to validate data produced by another system. The data is transferred to diskette, and a DE/RPG editing program can be used in rerun mode. Since the keyboard input is bypassed, it really doesn't matter whether the transaction file was produced by an operator or a system.

User Command Keys

When a format is executed from calcs via the EXFMT op code, the function and command keys listed in the I/O Operations section of the DE/RPG Reference Manual can be tested by the user program and treated as program function keys. An EXFMT is essentially a "write display" followed by a "read keyboard." Any of the listed keys will terminate the read and cause the statement following the EXFMT to be executed. If the STATUS keyword is specified for the job, any time an indicator is coded in columns 56-57, system error handling is suppressed and the user can process the error.

In the case of EXFMT, the status variable will contain the value (9501-9525) corresponding to the key depressed by the operator. The user can test for specific keys and take the appropriate action.

Output Fields

Output fields are processed twice for each display. First, when the screen is first displayed, all output fields are shown. Then all the fields are reprocessed in order of occurrence on the A specifications.

When the system displays unnamed output fields, the literal data displayed can never change. However, if the field is named, the initial value of the named field is placed on the screen. When that field is encountered the second time, the field's current value is displayed. If keyboard activity or calculations have modified the initial value, the modified value is displayed on the second pass.

SUBST Keyword

The nature of the substitute function places certain characteristics on the keyword's operation. Normally, the keyed data is found in one table, and a corresponding entry from a second table is placed in the field. However, if the field already contains a value, and new data is not keyed, no substitution occurs; ie, the substitution value is assumed to already be there. If the field is skipped, it will be blanked out by the 5280; thus blank is not valid data for a value in the second table.

If the index is coded, it is set when the substitution occurs. If the substitution is not performed, as in the situation mentioned, the index is not changed from its last value. Thus, if the program requires the proper setting of the index under all conditions, CHECK(ME) should also be coded to force reentry of the data each time the field is processed.

SEARCH Errors

When searching by record number, it is possible to request a record which has been deleted (3706 error). Searching by content or sequential content for a nonexistent record will produce a record not found error (3702 error). In all cases, pressing RESET will display the next nondeleted record in the transaction file. In the case of search content, this will be either the last or first nondeleted record in the file, depending on whether the search was in the forward or backward direction.

Edits on Bypass Fields

If a field is coded as unconditional bypass, no editing functions are performed on the data. If the field is conditional bypass, and the bypass indicator is off, the specified edits will be performed.

Edits on Duplicate/Skip Fields

All edits are performed when fields are duplicated or skipped. However, the error conditions associated with certain keywords (COMP, LOOK, RANGE, RANGET, SEQ, SUBST and XCHK) are bypassed. The assumption is that this data was manually entered previously and all such errors were caught at that time. If the original data was marked as being in error, all the duplicated fields are also marked. Later, the original error can be corrected and rerun mode used to correct all the following fields.

DE/RPG--EXPLANATIONS

The items in this section answer some frequently asked questions about DE/RPG capabilities.

Multivolume Files

DE/RPG will support multivolume files. Sequential files can be either one volume on line at a time, in which case the operator is prompted as new volumes are needed, or all volumes on line at one time. In the former case, only sequential op codes are allowed; if all volumes are online, then relative record number access is allowed.

Keyed files can be multivolume only if the file is built from scratch by a DE/RPG program and all volumes are online. If the Sort Key/Index option is used to build an index data set, the file is restricted to a single volume, since Sort does not support multiple volumes.

The online characteristics are specified on the file line of the A specification as follows:

Single volume at a time--DEVICE (DISK D1 D1)
All volumes online--DEVICE (DISK D1 D2...)

Arrays

Data entry tables can be accessed as arrays in calcs, but only one element at a time. That is, MOVEA and op codes that operate on array names only are not supported, but specification of the form AR,X is supported, where AR is the array name and X is an index (constant or variable). Thus, "MOVEL STATES,5 DEST" would move the fifth entry in the STATES array to the DEST field left justified.

The form "X,Y" is valid only on the C specifications, not on the A specifications. Array elements must be moved to/from a field (such as "DEST") to be used on the A specifications.

Functions Available Without a Transaction File

Most functions of DE/RPG are available even though no TFILE is specified, except for those which are operator controlled (such as searches). Format names can be specified on the Z specifications with all normal chaining, repeat count, conditional chaining and so on, as long as the WRITE(*NO) keyword is specified on the E line for the format. All editing functions are available to the format, including MARK FIELD and EDIT RELEASE. The only exception is DUP, which is not allowed because there is no previous record from which to duplicate.

The difference at job start time is that no mode prompt is displayed, and at EOJ no statistics prompt is displayed, since these are both functions that require a TFILE.

Object Program Storage Size

The program length on the compiler listing is not the size of the partition required to execute the object program. Physical buffer space for the diskette files must be added to the program size parameter to determine partition size. The physical buffers are a function of record length, exchange type and physical sector size of the diskettes involved at execution time. Thus a program that fits in a partition when using 256 byte physical records may not fit if 1024 byte physical records are used.

In general, a file that is Basic or H exchange requires a single physical buffer of 128 or 256 bytes, respectively. An I exchange dataset will need a physical buffer that is twice the size of the physical sector size (eg, 512 byte physical buffer for 256 byte physical sector), unless the record length is an even submultiple of the physical sector (eg, record length of 2, 4, 8, 16, 32, 64, 128, or 256 bytes for a 256 byte physical sector). In this case, a single physical buffer equal to physical sector is adequate.

If the record length is greater than the physical sector size, the physical buffer must be large enough to hold the minimum number of physical sectors required to ensure that one complete logical record is always in storage. For example, a 300 byte record on a 256 byte physical sector will need a physical buffer equal to three physical sectors, or 768 bytes, since the worst case record could span three physical sectors.

A transaction file must always be double buffered for the record insert function to work. Therefore, whatever size is determined above must be doubled for a TFILE.

The above calculations must be made for each file according to its execution time characteristics and the sum added to the compiler size parameter to determine the true partition size for the object program.

Usage Field

The usage field on the A specification determines when data will be displayed on the screen and what the source of the data is. The table below summarizes the action of the usage field.

Format					When Displayed	
Type	Mode	Usage	Source	Destination	Initially	2nd Pass
TFILE	Enter	I	Blanks	Name and Disk	No	Yes
		B	Blanks	Name and Disk	No	Yes
	Update	I	Disk	Name and Disk	Yes	Yes
		B	Disk	Name and Disk	Yes	Yes
EXFMT	Execute	I	Blanks	Name	No	Yes
		B	Name	Name	Yes	Yes
All	All	O	Literal or Name		Yes	Yes
		W	Insert Value	Name	No	No

Table Size

The value of NUMENT on a table description is the maximum number of entries allowed. However, if there are fewer entries in the diskette file or following the **CTDATA statement than the value in NUMENT, the actual table size at execution time is limited to the true number of entries. The excess space is not available for use by the program. Use dummy entries to fill out the table to its maximum size if you want the space.

Conversely, if the number of actual entries exceeds the value in NUMENT, table loading stops when the NUMENT value is reached. The excess entries are not available.

If there are no entries following the **CTDATA statement, the maximum table space in NUMENT is allocated and initialized to blanks if it is a character table or zeroes if it is a numeric table (decimal positions value is 0-9, maximum entry size is 15 bytes). **CTDATA is required for all compile timetables regardless of whether any data follows. Otherwise, a null table will result.

Marked Fields

A marked field contains an X'FF' in the leftmost position of the field. If the MARK keyword is used, an 'E' is placed in the specified position if one or more fields in the record are marked. The 'E' or X'FF' can be interrogated by the operator or a system to determine if the record is valid. If the field is named, the marked value goes into that storage location so the user can determine if good or bad data is being used.

When the fields in a record are corrected, the 'E' must be manually removed from the record.

Numeric Fields

Fields which are defined as numeric data (decimal positions 0-9) are carried in storage with F zones. Fields which are input as blanks are thus converted to zeroes. If such fields are subsequently displayed using the INSERT or AUXDUP keywords, or written to disk using the WRITE or UPDATE op codes, the value will appear as zeroes, not blanks. If they are written to the printer, the appropriate zero suppression codes would be required in the EDTCDE keyword.

Production Statistics

Production statistics values are available to the user program as read only fields accessed as *STAT1 to *STAT29. Details are in the Production Statistics section of the DE/RPG Reference Manual.

RPG to DE/RPG Conversion

The following table shows what RPG specification entries are used in DE/RPG and what corresponding 5280 specification entries are used. Unused RPG entries are not shown.

RPG		5280				
Form	Columns	Form	Name Type	Columns	Keyword	Comment
All	1-6	All		1-6		
H	20	Z	J	55-80	EDITC	
	21	Z	J	55-80	EDITC	
	23-25	A	F	30-34		
	26			1-17		**ALTSEQ
	43			1-17		**FTRANS
F	7-14	A	F	19-26		
	15					Implicit
	16	A	T	17		Tables Only
	20-23					Implicit
	24-27	A	F	30-34		
	28					Implicit
	29-30	A	b̸	30-34		
	31	A	K	17		Keyed Files Only
	35-38	A	b̸	30-34		
	40-46	A	F	45-80	Device	
	47-52	A	F	45-80	Device	
	60-65	A	F	45-80	Index	
E	11-18	A	F	19-26		
	27-32	A	T	19-24		
	33-35					Implicit
	36-39	A	F	45-80	Nument	
	40-42	A	T	30-34		
	44	A	T	37		
	46-51	A	T	19-26		
	52-54	A	T	30-34		
	56	A	T	37		
I	7-14	A	F	19-26		
	19-20	A	F	45-80	SETON	
	21-24	A	F	45-80	RECID	
	27	A	F	45-80	RECID	
	44-47	A	b̸	39-44		
	48-51	A	b̸	30-34		Length
	52	A	b̸	37		
	53-58	A	b̸	19-24		
O	7-14	A	F	19-26		
	17	A	R	45-80	SPACEB	
	18	A	R	45-80	SPACEA	
	19-20	A	R	45-80	SKIPB	
	21-22	A	R	45-80	SKIPA	
	32-37	A	b̸	19-24		
	38	A	b̸	45-80	EDTCDE	
	40-43	A	b̸	39-44		Starting Position

DE/RPG I/O OPERATION SUMMARY

The table below summarizes the coding required for the RPG I/O operations.

Operation	Factor 1	Op Code	Factor 2	Result	High Ind	Low Ind	Equal Ind
CHAIN	Rec# Key	CHAIN	Filename Recname		RNF	ERR[1]	
CLOSE		CLOSE	Filename			ERR	
DELETE	Blank Key	DELET	Filename		RNF	ERR	
EXFMT		EXFMT	Recname			ERR[2]	
OPEN		OPEN	Filename			ERR	
READ		READ	Filename Recname			ERR	EOF
READP		READP	Filename Recname			ERR	EOF
SETLL	Rec# Key	SETLL	Filename		RNF	ERR	RF
UPDATE		UPDAT	Recname			ERR	
WRITE		WRITE	Recname			ERR	EOF

EOF = End of File (underscore means required)
ERR = Device Error
1 = STATUS variable will contain error code
2 = Command Key, Function Key, or Record Backspace pressed
STATUS variable will contain key code
RF = Record found
RNF = Record not found (underscore means required)

Select Format

Any time a record is displayed with the wrong format, the correct format can be manually selected using the SEL FMT function. This function rereads the record from diskette when the new format is selected, so any data that may have been changed by a duplicate or skip in the wrong format is recovered in its original form.

PMT Keyword

The prompting keyword (PMT) is used to display a message on line 1 of the display only when the specific field is being entered. This message can be up to 200 characters long, but is generally less than one line. When the keyword is used, the first line of the display will always be cleared completely after the field is entered regardless of the message length. If the message is 81-200 characters long, the exact message length will be cleared.

Display Line Numbering

Although the displays are 480, 960 or 1920 characters in size, the status line always occupies the first physical line. Thus the size available to the DE/RPG user is 400, 880 or 1840. To facilitate user coding, DE/RPG allows line numbering for 1-5, 1-11 or 1-23 lines respectively. Because of the PMT keyword message placement, if no line is specified, the first field will start on user line 2. If you want fields on user line 1, you must specifically code line and position information on the A specification. This will result in a compiler warning (1068), which can be ignored. However, if the PMT keyword is used on a field following a field defined on line 1, the data on line 1 will be overlayed by the message.

Storage Updating

Named fields are updated in storage as soon as the data is entered. Thus a subroutine can use the last value entered for a field without the need to wait for the whole record to be completed.

DE/RPG TECHNIQUES

This section describes some coding and operational practices that will enhance the 5280's operation.

Compile Time Reduction

Compile time will be kept to a minimum if the following are done:

1 Use a 16K partition (little improvement beyond 16K).

2 Put SYSUTOX1 and SYSUTOX2 on separate drives.

3 Use 2D diskettes for workfiles.

4 Preallocate workfiles.

5 List to diskette rather than printer, then use the SYSPRINT utility to print the listing. Put the listing file on its own drive or on the same drive as SYSUTOX2.

6 Use a large physical sector size when initializing the diskettes on which the compiler and workfiles will be located. However, this will also increase the partition size required.

Multiple Access Paths in a Single Program

Keyed files may have multiple access paths into a single master file by using different index files. However, since only one INDEX keyword is allowed per file, and a file may not be opened twice by the same name, some special techniques must be followed.

1 Use the real file name with the first index.

2 Use dummy file names with subsequent indexes, and use the LABEL keyword with the real file name.

3 Specify SHARER or SHARE as required, using the real file name and all the dummy names.

4 The WRITE op code is not allowed for the file.

Multiple Index Data Sets

When a file is not shared, new records can be added to it. In the case of an indexed data set, the index file is updated immediately, so the new records are always available for access right away. There is no overflow area--a space is created in the index data set and the new key and relative record number are inserted into the proper place.

However, if other access paths exist, they are not updated and hence do not reflect any new records that they did not add. The other indexes must be rebuilt in order to access all the records.

Multiple Identical Keys

Multiple identical keys are allowed for a file, but DE/RPG does not support them. This means that when the search for the record (which is a binary search) is performed, the returned record could be the first, last or intermediate record with that key.

If multiple identical keys exist, the user must code READP and compare logic to find the first record with the key. Then READ and compare logic must be used to know when the last record with the key has been processed.

When using the DELET op code, this logic must be carefully followed or the wrong record could be deleted by mistake.

Shared Files

Programs may share files. The compiler implicitly determines the appropriate open to use (input/output/both) for each file in a program by analysis of the op codes used for the file. The SHARER and SHARE keywords are used to specify what operations one program will allow other programs to perform on the shared files as follows:

SHARER--Other programs may read, but not write.
SHARE--Other programs may both read and write.

A file that is shared for both reading and writing may not have insertions made to it by any program. A shared read/write sequential data set (ie, with no keys) may have new records added to it. However, a shared read/write keyed data set may not have records added to it because such a function would usually result in an insert function of some type, either to the data set itself or to the index, which is not allowed. In addition, when such a file is read by a program that both reads and writes the file, the sector containing the record read is locked until the next activity by the program on that file. Thus, depending on exchange type and record length, more than one record may be protected during concurrent update. If only some records are updated, an UPDAT op code should be used immediately following the read of a record which will not be updated in order to release the sector for access by other programs.

If two programs sharing the same file for both reading and writing are loaded from the same station into different partitions, a partition lockup is very probable, since one program may own a sector that contains a record the other program wants and vice versa. When this happens, an IPL is required to recover the partitions.

Variable Messages

Variable messages can be displayed using named output fields. Since output fields are processed twice, the initial display will show the initial value of the named location. As processing occurs, other messages may be moved into that location and displayed each time that named output field is encountered. Since the same display positions are reused, the compiler will generate a warning message (1065) which can be ignored.

As a practical matter, the message area should be blanked initially, then blanked again when the message is no longer required. The messages themselves can be created using the INSERT keyword to initialize working storage with literal constants. The format which includes these inserts must be executed for the initialization to take place. This can be done by defining a screen record with usage=W, making it the first format on the Z specification with a repeat count=1, and then chaining to the first data entry format or subroutine in the program after the working storage format has been completed.

The DSPATR keyword can be used when the named output fields are specified in a format to highlight the message with blinking or high intensity. Reverse image, underline or column separators should not be specified since they would be active even on a blanked message area.

CRT Editing

The EDTCDE keyword can be used for the CRT only when the entire screen record is output only. The WRITE op code is used with the desired record name. EDTCDE cannot be used if the format contains any I, B or W usage fields.

The subroutine which contains the WRITE op code should return to the Z specification and chain to an input format with the CLRL(*NO) keyword specified. Otherwise, the output screen will appear so briefly the operator will not have time to view it since processing continues as soon as the WRITE operation is complete with no operator response required or allowed.

*RTN Usage

A special field name, *RTN, is provided to force certain keyword functions and any user subroutines to be executed independent of any field data entry. *RTN is coded as a field name on the A specification, and the only other entries allowed are the keywords SETON, SETOF, RESET and EXSR. This function is useful for initializing fields and indicators prior to format execution.

Embedded Display Attribute Bytes

The hexadecimal codes for the display attributes can be included as part of a literal constant for an output-only field (usage=0) or a PMT keyword message. The hex values can be found in the Key Entry Utility chapter of the Utilities Reference/Operation Manual. The values are entered using the HEX key on the keyboard. On the display, SYSSEP will show these characters as a blob, and positions 18-19 of the status line will show the actual hex value when the cursor is positioned under the blob.

When using embedded attributes, the compile list to disk option must be used. The attribute codes are treated as SCS printer control by the 5256 and 5225 printers, causing strange printer results. Use the SYSPRINT utility to print the listing since it changes all nonprintable characters to "-" and thus eliminates the problem.

Review Second Data Set Function

This function is useful when entering new records into a transaction file, especially source programs. It is used to copy one or more records from an already existing data set to the transaction file. Use it for reducing the keying time for blocks of source code that you want to copy, data for compile time tables and self-check algorithm parameters. If records are copied one at a time, the operator can change the data before writing it to the transaction file without modifying the original records.

Chaining and User Program Control

The logical format chaining functions of the Z specification can be used even when EXFMT is used to control the keyboard/display and no TFILE is specified. The subroutine can store the desired chaining control character in a field and then return to the Z specification and chain to a format; eg, DUMMY, which has the keyword WRITE(*NO) coded. DUMMY would contain a one-character field with the INSERT keyword, using the control character field as the parameter. If the AUTO ENTER switch is on, the DUMMY format will barely be noticeable. On the Z specification, DUMMY would be coded to test position 1 for a value and chain to the appropriate next format or subroutine.

ENTRATR Keyword

This keyword is used on the Z specification (job line) when a desired set of attributes is to be used for all input fields in the program only when the cursor is within the field. Otherwise, the field is displayed with the normal attribute.

Position Field

The A specification does not require an entry in starting position for a field. Unless otherwise specified, the first field is assumed to start in position 1 of the record and all following fields are contiguous in the record.

In the case of display records, the assumed starting position is line 2, position 1. All following fields are assumed to be contiguous with no spaces between fields. A space is only forced if the DSPATR or ENTATR keywords have been coded, and then only on the fields to which they apply.

Previous Record Buffer Contents

When any operator function causes a different record to be displayed (ENTER, HOME, searches, etc), the previous record buffer is automatically loaded with the proper record to ensure correct operation of the duplicate function.

The Diskette Listing as Compiler Input

DE/RPG can operate on a printerless system. The list to diskette option produces 80-character print records. SYSSEP can be used to scan the diskette listing for the error messages, the corrections made in the listing file, and the corrected listing file used as SYSIN for the compiler. The compiler-generated messages have a special code in Column 7 which displays as a blob and causes the compiler to ignore them on input. If SYSPRINT is used to print the diskette listing, it will convert the special character to a "-" prior to printing it.

Debugging Functions

DE/RPG debugging aids are contained in two manuals:

> DE/RPG Problem Determination Procedures for the Programmer, SC21-7852
>
> Data Areas and Diagnostic Aids Handbook, SY31-0595, Section 6 (Diagnostic Aids)

UTILITY NOTES

This section provides some explanations of certain utility operations that will be helpful during system operation.

SYSCLOSE

SYSCLOSE is intended to aid recovery of data from files when an abnormal termination results in failure to update the data set header label. This is generally the result of a power failure but may also occur if a re-IPL is necessary and a program is not terminated by the operator prior to the IPL. It may also happen if the operator incorrectly selects ENTER mode on an existing file and is presented with the first record before recognizing the mistake.

The utility simply opens the file and sets the End of Data equal to the End of Extent. The operator searches the data set for the last recognizable valid record, then presses FIELD EXIT and ENTER. The End of Data is set to the correct record and the file is closed. It may then be properly used by other programs again.

SYSCOPY

The image Copy option of SYSCOPY requires that both the FROM and TO diskettes be exactly alike (volume labels must be identical as far as physical description parameters are concerned) and that the TO diskette be completely empty. SYSCLEAR can be used to remove all data sets from the TO diskette; SYSINIT can be used to clear the TO diskette and create the proper volume label parameters. If the TO diskette is not the same as the FROM diskette, use the Volume Copy to copy to the TO diskette.

If Data Set Copy is used, and the TO data set is not preallocated, the copy will attempt automatic allocation of the exact same exchange type data set as the FROM data set. If this cannot be done, a data set type mismatch error (3218 or 3236) occurs. Use SYSLABEL to allocate the TO data set, then rerun the Data Set Copy.

Data Set Copy allows up to four different FROM data sets to be concatenated into a single TO data set. It also permits inclusion of deleted and marked records, which is useful when copying source programs. If all the data cannot fit on the volume, a multivolume support option is allowed (if not taken, the TO data set is cleared if it was preallocated or deleted if it was allocated by SYSCOPY). In the case of an existing TO data set, the Data Set Copy can either add to or overwrite the TO data set.

SYSKEU

SYSKEU is intended only for the simplest nonintelligent key entry functions, and then only as a migration tool for 3740 users. It is not as functional as the 3740 in the key entry sense, but does include many 5280 ease-of-use (eg, SEARCH prompts, etc) and statistics functions.

The SYS3740C utility can be used to convert existing 3740 key entry program levels to DE/RPG source Z and A specifications. These can then be compiled directly into DE/RPG object programs, or enhanced with 5280 intelligent functions using SYSSEP prior to compilation.

SYSLABEL

When SYSLABEL (or SYSCLEAR) is used to free (delete) a data set, the physical area on the diskette occupied by that file is not available for future use without running SYSCOMP to eliminate unused space unless the file was physically the last one on the diskette. When SYSLABEL allocates a file, it uses the first deleted header record available for the extent information but physically locates the file at the next record following the last physical file. Thus, if the first header had a beginning of extent at track 1 sector 1 and that file was subsequently deleted, the next file allocated would appear as the first header but its track and sector BOE would be the highest of all the files defined.

SYSSORT

SORT must allocate its own output files for the command data set and output. Thus if a replacement command data set is created, or a rerun of the sort is to use the same named output file, the existing command data set or output file must be deleted using the Data Set Free option of SYSLABEL (delete function) or SYSCLEAR.

SYSACLC

This utility is intended to assist in converting 3741 ACL source programs to 5280 Assembler source. Some batch programs may convert 100%, but generally some additional Assembler statements will be required.

The utility generates a relatively large amount of control code which is used to reconcile differences between the 3741 and the 5280, thus producing output that will operate very much like the 3741 program. Depending on the 3741 functions used, the 5280 assembled object will range from about 50-150% larger than the 3741 object; ie, an 8K ACL program will take 12-20K on the 5280.

If this is not acceptable, it would be better to recode the application in DE/RPG than in 5280 Assembler because of programmer productivity.

GENERAL NOTES

5280 COBOL Considerations

5280 COBOL is based on the Series/1 program product. It does not support the sort or merge modules, but it does have the following additions:

- Work station support
- Communication support (via a CALL interface)
- Complex conditions
- Nested IFs
- Nested REDEFINEs

5280 COBOL is compiled on a host system and executed on the 5280. It is record oriented and does not have any IBM-supplied intelligent data entry edits--the user codes any field validation tests (eg, table look-up, range check, etc) in his own program. These tests are performed after the record is completely keyed by the operator.

5280 COBOL uses the A specifications to supply screen display attributes for fields and to specify those edits that are performed by the keyboard/display hardware (data type, field position and length, and some CHECK keyword parameters). These are defined externally to the COBOL source program and copied in as needed. Thus, a record need only be defined once for the system, yet can be used by any COBOL source program.

5280 COBOL displays are program controlled (as are those used by the EXFMT op code in DE/RPG). The various function and command keys are treated as program function keys. The user includes in his program any processing desired for those keys which the operator is to be allowed to use.

It should be noted that while 5280 COBOL display management conforms to the GSD standard, there is no external (outside IBM) standard for interactive device management. The CODASYL Screen Management Task Group is working on such a standard, but its direction and availability are not firm. The 5280 implementation represents the proper step for 5280 COBOL at this time.

5280 Assembler Considerations

The 5280 Assembler is provided primarily to allow minor modifications to 3741 ACL source programs which have been converted by the SYSACLC utility to 5280 Assembler source. It is a very powerful assembler, but there is no formal education available for it.

The user is responsible for coding everything in the Assembler, including what is normally considered to be operating system function. There is no operating system on the 5280. The common function area consists of system tables, error handling routines, common system routines, and common DE/RPG routines only. (The DE/RPG routines are contained in SYSCFA or SYSHELP only.) The user may link to most of these functions, but no access is allowed by user code to any of the common DE/RPG routines in order to ensure system integrity.

The user will have to write special code in Assembler if more than one partition is to operate successfully in a manner similar to normal 5280 function as provided by the IBM software.

The user should very carefully analyze the manpower requirement, development time, and development cost of using the Assembler to save storage. The high productivity of DE/RPG and COBOL and the low cost of storage on the 5280 will probably offset any potential memory and/or dollar savings that might accrue from using the Assembler.

Partition Concepts

When using several utilities together, it is faster and easier to load them into different partitions all at once, and then select the one desired as needed. To load another partition, press the SYS REQ key at any time and the load prompt will be displayed. Enter the program name and address manually, but include a partition number. The specified program will be loaded into that partition and begin execution.

Press ATTN to switch between active partitions loaded by your station.

Background partition loading and switching between partitions is described in 5280 System Concepts (GA21-9352), Chapter 5.

Cancelling a Runaway Background

A station can be forced to attach to any background partition loaded from that station by using the console mode. Press CMD and C for console mode, then B to attach to a background followed by the background partition number. To end the job, press EOJ (some communication utilities will not end at this point; in that case, press CMD and CNCL).

The Console Mode as described in 5280 System Concepts (GA21-9352), Chapter 5.

Resource Allocation Table

The resource allocation table is used to free programs from a dependence on any particular physical device addresses. It takes the place of device information contained in JCL or OCL, neither of which is used on the 5280.

If we assume a two-station system with disk address 4400 on one station and 4800 on the other station and a disk address 4000 somewhere on the system, the following operation is possible.

A program diskette could always be resident on 4000 and a resource allocation table could be built using 4400 as logical D1 on one station and 4800 as logical D1 on the other station. The programmer can then code a file device address as D1. Whichever station then loads the program will always be requested to place that file on the drive directly in front of the operator, since D1 will be converted via the resource allocation table to either 4400 or 4800. These addresses can be overriden at execution time if necessary (except for SORT/MERGE), but provide excellent ease of use for the operator/programmer interface.

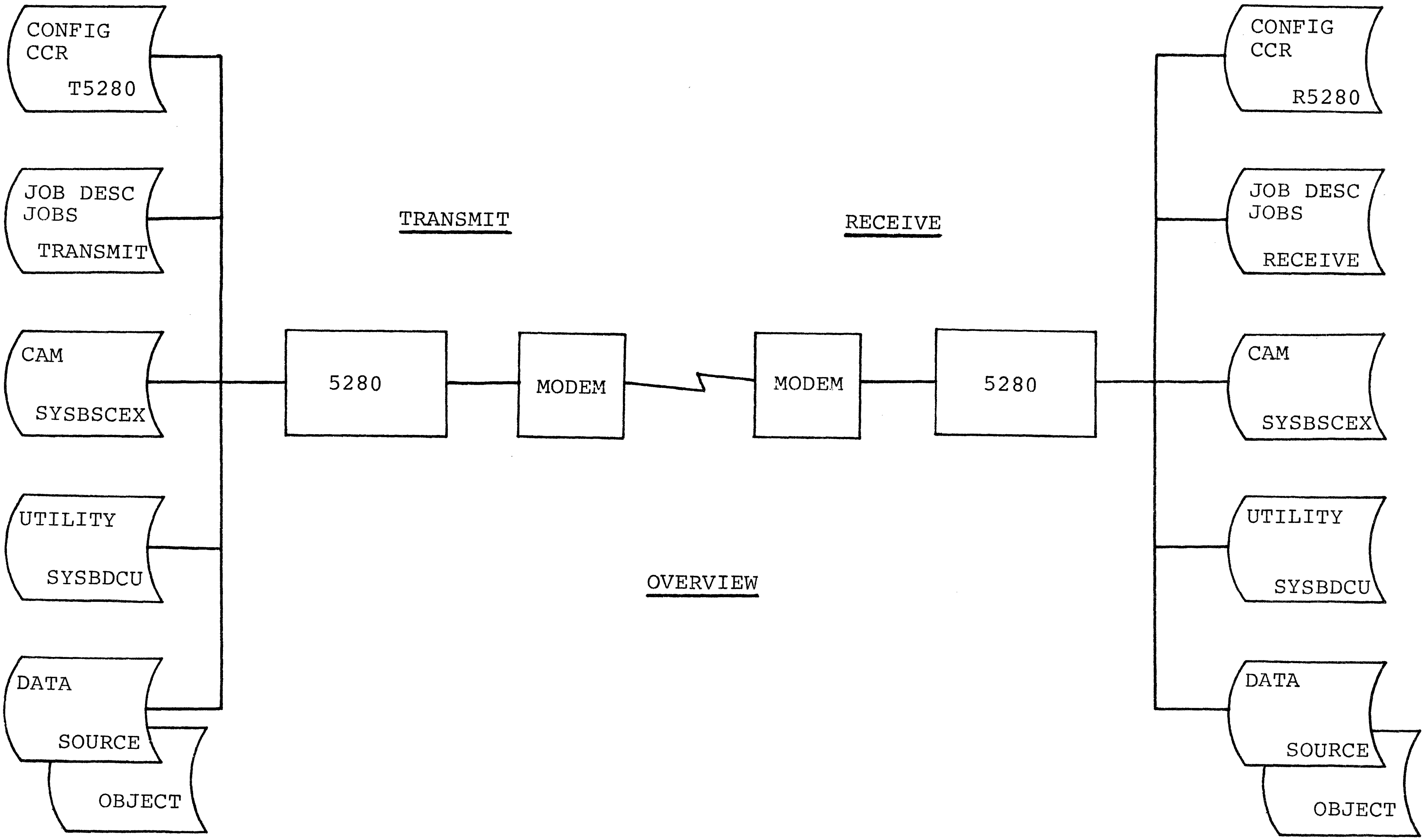
5280 to 5280 Transmission

The following pages show the required communications records necessary to send a source and an object program between two 5280 locations. The best procedure is to compile using the list to diskette option, with SOURCE as the listing file. SOURCE is the source program data set. On the same diskette, copy the object program to a file called OBJECT.

Load this diskette into drive 4000, then proceed with communications. The receiving location should load a diskette into 4000 which already has the two data sets allocated (SOURCE should have about 1000 records of length 80; OBJECT should have about 128 records of length 256; these must be large enough to hold the data sets being received).

This will be useful in sending programs between the branch and region or between branches, or to Rochester when necessary. In the case of programming problems, the remote location can quickly get a copy of the source and object, then step through the program simultaneously with the sending site to analyze the problem.

Using a 2400 bps line, the transmission rate will be about 96 source statements per minute or 13K of object code per minute.


CONFIG
CCR
T5280
JOB DESC
JOBS
TRANSMIT
CAM
SYSBSCEX
UTILITY
SYSBDCU
DATA
SOURCE
OBJECT
TRANSMIT
RECEIVE
5280
MODEM
MODEM
5280
CONFIG
CCR
R5280
JOB DESC
JOBS
RECEIVE
CAM
SYSBSCEX
UTILITY
SYSBDCU
DATA
SOURCE
OBJECT

OVERVIEW

```
             5280 COMMUNICATIONS CONFIGURATION

DATASET NAME  -       CCR             RECORD NAME   -     T5280

NETWORK TYPE          BSC             CLOCKING              MODEM
WRITE PROTECT         NO              MODEM RATE            FULL
LINE TYPE             SWITCHED        CON REQ-TO-SEND       NO
LINE INTERFACE        EIA             DSR TIMEOUT           01
INT MODEM ATTEN                       LINE SPEED            2400
SNBU                  NO              CLR-TO-SEND DLY       01
CONNECT TYPE          MAN             CAM NAME              SYSBSCEX
CONNECT DS-TO-LINE    NO              LINE CODE             EBCDIC
ANSWER TONE ORIGIN    BUS MACH        TRACE TABLE ENTRIES   000
ANSWER TONE TIMER                     LEADING PADS          03
ANSWER TONE FREQ                      PAD CHARACTER         55
WRAP MODEM            NO              UNDERRUN RETRY        5


                    - - - - -  BSC  - - - - -

DEVICE TYPE           3741            RVI                   NO
INIT LINE BID         5280            TTD RECV LIMIT        07
PRIMARY STA           YES             CAM BUFFER            00514
TERMINAL ID                           LINE BID RETRY MAX    99
HOST ID                               EOTS TO EOT           00
POLL ADDRESS                          TRAILING PADS         03
SELECT ADDRESS                        RECV TIMEOUT          27
RECORD FORMAT         UNBLOCK         RECV T/O RETRY        99
ITB BLK FACTOR                        XMIT TIMEOUT          02
IRS                   NO              CONTINUE T/O          15
BLOCK SIZE                            WACK RECV LIMIT       015
TRANSPARENCY          YES             RECORD XMIT LIM       02
BLANK EXPANSION       NO              CHAR SYN COUNT        2
BLANK TRUNCATION      NO
```

Transparency is required to transmit object data sets.

TRANSMIT CONFIGURATION RECORD

Actual responses to SYSCCU prompts when creating this configuration record.

```
CCR
T5280
BSC

* C1   2
* C2   2
  C3
  C4
* C5   1
  C6
  C7
  C8
  C9
* C10  2
* C11  2
* C12  1
  C13
* C14  01
* C15  2400
* C16  01
* C17  SYSBSCEX
* C18  1
* C19  000
* B1   1
* B2   1
* B3   1
* B4
* B5
  B6
  B7
* B8   1
  B9
  B10
  B11
* B12  1
  B13  2
* B14  2
* B15  2
* B16  07
* B17  00514
* B18  99
* B19  00
* B20  03
* B21  03
* B22  27
* B23  99
* B24  02
* B25  15
* B26  015
* B27  02
   ↑   ↑
   |   Actual response
   |
  Question ID
```

5280 COMMUNICATIONS JOB DESCRIPTION

TRANSMIT JOB AND FUNCTION DESCRIPTION RECORDS

JDR	JOB NAME		TRANSMIT
02	UTILITY/PROTOCOL	1	SYSBDCU 3741
03	CONFIG DATASET		CCR
03	CONFIG ADDRESS		D1
03	CONFIG RECORD		T5280
03	CAM PARTITION		①
04	MODE	1	ATTEND
05	LOGON(SDLC)		
06	FIRST FUNCTION		TSOURCE
03	CONFIG DEVICE ADDR	2	LOGICAL

FDR	FUNCTION NAME		TSOURCE
08	FUNCTION OPTION	1	TRANSMIT
09	DEVICE OPTION XMIT	1	DISKETTE DS
10	PRINTER DATA 10/11		
12	CONSOLE DATA OPT		
13	DATASET (XMIT)		SOURCE
13	DATASET (PRINT)		
13	DATASET (CONSOLE)		
13	DEVICE ADDR T/R/PU		D1
15	DEVICE ADDR (PRT)		
13	DEVICE ADDR (CONS)		
14	PCH OR 3741 RECV		
14	WRITE PRINT DATA		
14	WRITE CONSOLE		
15	PRINT ADDRESS		
16	PRINT FMT 3741 RCV		
17	PRINT FMT NAME		
22	TRANS ID OPTION	1	NONE
22	TRANS ID		
24	ALT FUNCTION OPT	2	NO
24	ALT FUNCTION NAME		
25	NEXT FUNCTION OPT	2	MORE
25	NEXT FUNCTION NAME		TOBJECT
02	UTIL/PROTOCOL OPT	1	SYSBDCU 3741
04	MODE	1	ATTEND
13	PCH DEVICE (MULTI)		1
13	PRT DEVICE (MULTI)		
13	CONSOLE DEVICE	1	PHYSICAL
13	PRINT DEVICE	1	PHYSICAL

FDR	FUNCTION NAME		TOBJECT
08	FUNCTION OPTION	1	TRANSMIT
09	DEVICE OPTION XMIT	1	DISKETTE DS
10	PRINTER DATA 10/11		
12	CONSOLE DATA OPT		
13	DATASET (XMIT)		OBJECT
13	DATASET (PRINT)		
13	DATASET (CONSOLE)		
13	DEVICE ADDR T/R/PU		D1
15	DEVICE ADDR (PRT)		
13	DEVICE ADDR (CONS)		
14	PCH OR 3741 RECV		
14	WRITE PRINT DATA		
14	WRITE CONSOLE		
15	PRINT ADDRESS		
16	PRINT FMT 3741 RCV		
17	PRINT FMT NAME		
22	TRANS ID OPTION	1	NONE
22	TRANS ID		
24	ALT FUNCTION OPT	2	NO
24	ALT FUNCTION NAME		
25	NEXT FUNCTION OPT	1	LAST
25	NEXT FUNCTION NAME		
02	UTIL/PROTOCOL OPT	1	SYSBDCU 3741
04	MODE	1	ATTEND
13	PCH DEVICE (MULTI)		1
13	PRT DEVICE (MULTI)		
13	CONSOLE DEVICE	1	PHYSICAL
13	PRINT DEVICE	1	PHYSICAL

↑ Meaning or actual response

↑ Option selected

↑ 69-XX Screen number for this response

The configuration record created by SYSCCU is referenced in the job description by CONFIG DATASET, ADDRESS and RECORD.

The first function transmits the SOURCE data set.

The second function transmits the OBJECT data set.

○ The CAM partition is system IPL dependent.

```
          5280 COMMUNICATIONS CONFIGURATION

DATASET NAME  -       CCR              RECORD NAME   -     R5280

NETWORK TYPE          BSC              CLOCKING            MODEM
WRITE PROTECT         NO               MODEM RATE          FULL
LINE TYPE             SWITCHED         CON REQ-TO-SEND     NO
LINE INTERFACE        EIA              DSR TIMEOUT         01
INT MODEM ATTEN                        LINE SPEED          2400
SNBU                  NO               CLR-TO-SEND DLY     01
CONNECT TYPE          MAN              CAM NAME            SYSBSCEX
CONNECT DS-TO-LINE    NO               LINE CODE           EBCDIC
ANSWER TONE ORIGIN    BUS MACH         TRACE TABLE ENTRIES 000
ANSWER TONE TIMER                      LEADING PADS        03
ANSWER TONE FREQ                       PAD CHARACTER       55
WRAP MODEM            NO               UNDERRUN RETRY      5


                  - - - - -  BSC  - - - - -

DEVICE TYPE           3741             RVI                 NO
INIT LINE BID         HOST             TTD RECV LIMIT      07
PRIMARY STA           NO               CAM BUFFER          00514
TERMINAL ID                            LINE BID RETRY MAX  99
HOST ID                                EOTS TO EOT         00
POLL ADDRESS                           TRAILING PADS       03
SELECT ADDRESS                         RECV TIMEOUT        27
RECORD FORMAT         UNBLOCK          RECV T/O RETRY      99
ITB BLK FACTOR                         XMIT TIMEOUT        02
IRS                   NO               CONTINUE T/O        15
BLOCK SIZE                             WACK RECV LIMIT     015
TRANSPARENCY          YES              RECORD XMIT LIM     02
BLANK EXPANSION       NO               CHAR SYN COUNT      2
BLANK TRUNCATION      NO
```

Transparency is required to receive object data sets.

RECEIVE CONFIGURATION RECORD

Actual responses to SYSCCU prompts when creating this configuration record.

```
 CCR
 R5280
 BSC

* C1    2
* C2    2
  C3
  C4
* C5    1
  C6
  C7
  C8
  C9
* C10   2
* C11   2
* C12   1
  C13
* C14   01
* C15   2400
* C16   01
* C17   SYSBSCEX
* C18   1
* C19   000
* B1    1
* B2    2
* B3    2
* B4
* B5
  B6
  B7
* B8    1
  B9
  B10
  B11
* B12   1
  B13   2
* B14   2
* B15   2
* B16   07
* B17   00514
* B18   99
* B19   00
* B20   03
* B21   03
* B22   27
* B23   99
* B24   02
* B25   15
* B26   015
* B27   02
   ↑    ↑
   |    Actual response
   |
 Question ID
```

5280 COMMUNICATIONS JOB DESCRIPTION

RECEIVE JOB AND FUNCTION DESCRIPTION RECORDS

JDR	JOB NAME	RECEIVE
02	UTILITY/PROTOCOL	1 SYSBDCU 3741
03	CONFIG DATASET	CCR
03	CONFIG ADDRESS	D1
03	CONFIG RECORD	R5280
03	CAM PARTITION	1
04	MODE	1 ATTEND
05	LOGON(SDLC)	
06	FIRST FUNCTION	RSOURCE
03	CONFIG DEVICE ADDR	2 LOGICAL

FDR	FUNCTION NAME	RSOURCE
08	FUNCTION OPTION	2 RECEIVE
09	DEVICE OPTION XMIT	1 DISKETTE DS
10	PRINTER DATA 10/11	
12	CONSOLE DATA OPT	
13	DATASET (XMIT)	SOURCE
13	DATASET (PRINT)	
13	DATASET (CONSOLE)	
13	DEVICE ADDR T/R/PU	D1
15	DEVICE ADDR (PRT)	
13	DEVICE ADDR (CONS)	
14	PCH OR 3741 RECV	3 REPLACE
14	WRITE PRINT DATA	
14	WRITE CONSOLE	
15	PRINT ADDRESS	
16	PRINT FMT 3741 RCV	2 NO
17	PRINT FMT NAME	
22	TRANS ID OPTION	
22	TRANS ID	
24	ALT FUNCTION OPT	2 NO
24	ALT FUNCTION NAME	
25	NEXT FUNCTION OPT	2 MORE
25	NEXT FUNCTION NAME	ROBJECT
02	UTIL/PROTOCOL OPT	1 SYSBDCU 3741
04	MODE	1 ATTEND
13	PCH DEVICE (MULTI)	1
13	PRT DEVICE (MULTI)	
13	CONSOLE DEVICE	1 PHYSICAL
13	PRINT DEVICE	1 PHYSICAL

FDR	FUNCTION NAME	ROBJECT
08	FUNCTION OPTION	2 RECEIVE
09	DEVICE OPTION XMIT	1 DISKETTE DS
10	PRINTER DATA 10/11	
12	CONSOLE DATA OPT	
13	DATASET (XMIT)	OBJECT
13	DATASET (PRINT)	
13	DATASET (CONSOLE)	
13	DEVICE ADDR T/R/PU	D1
15	DEVICE ADDR (PRT)	
13	DEVICE ADDR (CONS)	
14	PCH OR 3741 RECV	3 REPLACE
14	WRITE PRINT DATA	
14	WRITE CONSOLE	
15	PRINT ADDRESS	
16	PRINT FMT 3741 RCV	2 NO
17	PRINT FMT NAME	
22	TRANS ID OPTION	
22	TRANS ID	
24	ALT FUNCTION OPT	2 NO
24	ALT FUNCTION NAME	
25	NEXT FUNCTION OPT	1 LAST
25	NEXT FUNCTION NAME	
02	UTIL/PROTOCOL OPT	1 SYSBDCU 3741
04	MODE	1 ATTEND
13	PCH DEVICE (MULTI)	1
13	PRT DEVICE (MULTI)	
13	CONSOLE DEVICE	1 PHYSICAL
13	PRINT DEVICE	1 PHYSICAL

Meaning or actual response

Option selected

69-XX Screen number for this response

The configuration record created by SYSCCU is referenced in the job description by CONFIG DATASET, ADDRESS and RECORD.

The first function receives the SOURCE data set.

The second function receives the OBJECT data set.

○ The CAM partition is system IPL dependent.

```
VOL1VOLID               IBM XXXX      OWNERID                                           101 W

Format  2

DATA SET NAME        EX BOE    EOD    EOE   WP    RECL     #REC     UNUSED

SYSCLU               I 01001 03006 03005         256        35          0
SYSBSCEX             I 03006 07007 07006         256        61          0
SYSBDCU              I 07007 13002 13001         256        85          0
CCR                  I 13002 13004 13010         256         9          7
JOBS                 I 13011 14002 14015         256        20         14
SYSLIST              I 15001 16011 16010         256        25          0
SYSPRINT             I 16011 19001 18015         256        35          0
SOURCE               I 19001 19001 39013          80      1001       1001
OBJECT               I 39014 39014 48006         256       128        128
```

SAMPLE DISKETTE DIRECTORY

Print Control Considerations

Many installations of communicating 3741 have a printer attached for offline printing of data received on diskette. This data can be printed by 1) a straight listing record-for-record, 2) formatted by 3741 format control, or 3) print control characters imbedded in the data stream. On 5280, Case 1 would be accomplished by using SYSPRINT utility, Case 2 by using DE/RPG program to read records and format the report, and Case 3 by using SYSPRINT or DE/RPG Program. Case 1 and 2 are straight-forward, but Case 3 deserves some special attention. The data must be received transparent to allow the hex print control to be passed to diskette. Additionally, the first record in the print data set must have the format command at the beginning of the record. The format command defines the page size and graphic error options. Other forms control commands (carriage return, line feed, next line, null, forms feed, bell, and IRS) function normally.

In an RJE environment, the SCS commands are intepreted by the CAM when directing output to the printer.

SCS Control Characters

SCS	HEX	FUNCTION
Bell	2F	Allow all preceding data to be processed, drop ready, attention on, sound alarm, stop print, stop formatting, unavailable to controller.
CR	OD	Return carriage to Position 1, same line.
FF	OC	Move paper to next page set by vertical FMT control.
FMT	2B	Defines form width, length (in positions), and error action. 2BC102hh where hh = positions (hex) 2BC202vv where vv = lines (hex) 2BC602LL where LL = LPI (hex) 5225 only 2BD2042900cc where cc = CPI (hex) 5225 only 2BC803gguu where gg = character error, uu = unit error
IRS	1E	Inter record separator: works like NL.
LF	25	Moves paper without altering print position, overflows if last line.
NL	15	Moves paper one line, returns carriage, overflows if last line.
NUL	00	No function performed.
PP	34	Set print position: Absolute or relative (34CXnn). 34COnn = Absolute horizontal move to Pos nn o < nn ≤ 132. 34C4nn = Absolute vertical move to Line nn. 34C8nn = Relative horizontal move to Pos nn. 344Cnn = Relative vertical move to Line nn. Values converted to absolute cannot exceed maximums in FMT command

See 5280 Functions Reference Manual (GA21-9353)--Appendix B for more information.

Print data containing SCS print control characters is supported by the 5280. Defaults for form width and length are record size and 66 lines respectively. SCS commands are available to set these values in the data stream. It is advisable to always issue these commands to insure proper forms environment. The first record in the data set should contain SCS commands for form length and width. If the 5225 printer is used, include lines-per-inch and characters-per-inch control if different from defaulted values. Use caution when accepting the form width default, since records shorter than your intended print line force a carriage return/line feed when printing at that rightmost position. Some suggestions:

1 Avoid printing in the last print position.
2 Keep records at least as long as the print line.
3 Avoid using print width default.
4 If SCS width control is used, any record length can be used without affecting automatic CR.
5 A DE/RPG program to print this data may be coded in two ways:
  A Use DEVICE(DISK) but address the printer. (See example, note I/O error action)
  B Use SPACEA(0) on all print record statements to suppress automatic forms control. However, a CR will be issued after each print line is printed.

If you are in a mixed 3741/5280 environment, it is important to know that the 3741 receives blocked print data to diskette different than the 5280. The 3741 receives the print data on diskette as it was received on the communication line. The 5280, however, expands the printer IGS control characters prior to writing them to diskette.

This means that you should use caution when printing data sets on the 5280 that were intended for the 3741.

Although the communication utilities support multivolume files, SYSPRINT does not have this provision. To print data received on a multivolume file, options are:

1a Allocate a data set large enough for all volumes.

1b Copy the data down to one volume with the data set copy function of SYSCOPY, using the "add to" option. Use SYSPRINT to print the file, or

2 Use a DE/RPG program to print the files. See following example. Be sure the record size matches the specification in your program.

EXAMPLE

```
ZJ   PRTSCS                                       STATUS (RETCD)
Z 1  CALCS          IE                            EOJ
 *Input File--SCS CHARACTER STRING

A         F SCSIN        256            DEVICE (DISK D1 D1)
A         R INREC            I
A           DATA         256 I          1
 *
 *Output File--DIRECTED TO A PRINTER BUT DESCRIBED TO DE/RPG AS DISK
 *             TO AVOID THE ADDITION OF MORE SCS CONTROL CHARACTERS
 *
A         F SCSOUT       256 O          DEVICE (DISK X'8000')
A         R OUTREC           O
A           DATA             O          1
 *
 *
 *CALCULATIONS TO DO THE PRINTING--2401 ERRORS ON THE OUTPUT FILE ARE
 *        IGNORED, SINCE THEY ARE A RESULT OF TREATING THE PRINTER AS A DISK
 *
C           CALCS   BEGSR
C           LOOP    TAG
C                   READ INREC                            10
C    N10            WRITEOUTREC                         20
C    N10 20 RETCD   CABEQ2401 LOOP
C    N10N20         GOTO LOOP
C                   ENDSR
```

Publication Orders

It is necessary to enter Licensed Program numbers when ordering 5280 publications using SLSS (Form Z120-1816) in order to ensure an accurate customer profile. If the Licensed Program numbers are not listed in Section E2, subject codes relating to those products will be ignored. The required Licensed Program codes are:

Licensed Program	Program Number
Assembler	5708-AS1
COBOL-OS/VS Host Compiler	5708-CB1
COBOL-DOS/VSE Host Compiler	5708-CB2
Communications	5708-DC1
DE/RPG Compiler	5708-DE1
System Control Program	5708-SC1
Sort/Merge	5708-SM1
Utilities	5708-UT1

Cable Orders

There is no ISR representative involved in 5280 orders. Therefore, the marketing representative is responsible for ordering 5280 cables if they are to be obtained from IBM. Since 5280 is a customer setup machine, the ISR has no responsibility for the account. The configurator asks if cables are to be ordered from IBM, but the response does not order the cables. Please ensure that you have handled this aspect of your 5280 orders.