



GC21-7728-4

File No. S38-00

IBM System/38

**IBM System/38
Introduction**



GC21-7728-4

File No. S38-00

IBM System/38

**IBM System/38
Introduction**

Fifth Edition (March 1982)

This is a major revision of, and obsoletes, GC21-7728-3. This edition applies to release 4, modification 0 of the IBM System/38 CPF (Program 5714-SS1), RPG III (Program 5714-RG1), COBOL (Program 5714-CB1), IDU (Program 5714-UT1), Conversion Reformat Utility (Program 5714-CV2), System/3 Batch Conversion Utilities (Program 5714-CV1), IBM System/3 CCP to System/38 Conversion Aid (Program 5714-CV7), IBM System/34 to System/38 Conversion Aid (Program 5714-CV5), Remote Job Entry Facility (Program 5714-RC1), and to all subsequent releases until otherwise indicated in new editions or technical newsletters. See *About This Manual* for a summary of major changes to this edition.

The functions and capabilities this publication describes reflect the latest information available, but this information may be subject to minor changes before release 4 of the IBM System/38 is available. Subsequent editions of this publication will reflect any changes.

In device operations, the nominal or rated throughputs represented in this publication may not be achieved in an actual customer environment when used with the customer's control program/application load.

Use this publication only for the purposes stated in *About This Manual*.

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

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

ELECTROMAGNETIC INTERFERENCE

United States Federal Communications Commission (FCC) rules require IBM to publish the following information, which applies to each unit (including modems and terminals) described in this manual (hereafter called *equipment*).

WARNING: *This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with the instruction manuals, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.*

Instructions Pursuant to FCC Part 15, Subpart J

1. This equipment should not be installed less than 30 meters (98.5 feet) from radio or television receivers or their receiving antennas.
2. This equipment should not receive its power from branch circuits that also power radio or television receivers.

If this equipment cannot comply with the above distance limitation:

1. Turn the equipment and the receiver on. If no interference is apparent, no further action need be taken.
2. If interference occurs, reorient or relocate the equipment, the receiver, or both.
3. If interference still occurs, contact your IBM representative.

Contents

ABOUT THIS MANUAL	vii	CHAPTER 3. MACHINE SUMMARY	3-1
Purpose of This Manual	vii	System Unit	3-3
Organization of This Manual	vii	High-Level Machine Interface	3-3
Summary of Changes	vii	Processing Unit	3-4
If You Need More Information	vii	Storage Management	3-6
CHAPTER 1. INTRODUCTION TO IBM SYSTEM/38	1-1	Virtual Address Translator	3-6
Program Products	1-3	Internal Input/Output Channel	3-6
Control Program Facility (CPF)	1-3	Main Storage	3-7
RPG III	1-7	Time-of-Day Clock	3-7
COBOL	1-8	Automatic Initial Microprogram Load	3-7
Interactive Data Base Utilities (IDU)	1-9	Power Keylock	3-7
Text Management Program RPQ	1-10	Power Warning Feature	3-8
Remote Job Entry Facility (RJEF)	1-10	Audible Alarm and Attention Indicator	4-8
System/3 Batch Conversion Utilities	1-10	System Console	3-8
Communications Control Program (CCP)		System Console Display Screen	3-9
Conversion Aid	1-10	System Console Keyboard	3-10
Conversion Reformat Utility	1-11	System Console Operator/Service Panel	3-11
System/34 to System/38 Conversion		Auxiliary Storage	3-13
Aid Licensed Program	1-11	System Unit Data Storage Function	3-13
Machine	1-12	3370 Direct Access Storage Device	3-14
System Unit	1-12	Diskette Magazine Drive	3-15
Input/Output Devices	1-13	Printers	3-17
CHAPTER 2. SYSTEM/38 PROGRAM PRODUCTS	2-1	5211 Printer	3-20
Control Program Facility (CPF)	2-2	3262 Printer	3-21
User Interfaces	2-4	3203 Printer	3-22
Control Language Programs	2-11	Card Device	3-23
Batch Input Streams	2-12	5424 Multi-Function Card Unit	3-23
Object Management	2-12	Magnetic Tape Device	3-24
Managing Work	2-14	3410/3411 Magnetic Tape Subsystem	3-24
Data Description Specifications (DDS)	2-19	Communications	3-25
Data Base Data Management	2-22	Work Station Controllers	3-26
Device Data Management	2-27	Communications Attachment	3-29
Programmer Services	2-31	CHAPTER 4. COMPATIBILITY/CONVERSION	4-1
Testing and Debugging	2-32	Compatibility with System/3 and System/34	4-1
System Operator Services	2-35	Conversion from System/3 Batch	4-2
System Services	2-36	Conversion from System/3 CCP	4-6
RPG III	2-41	Conversion from System/34	4-7
COBOL	2-47	GLOSSARY	G-1
Extensions Supporting Data Base and Work		INDEX	X-1
Station Processing	2-49		
Programmer Services	2-50		
Interactive Data Base Utilities (IDU)	2-52		
Source Entry Utility (SEU)	2-52		
Screen Design Aid (SDA)	2-53		
Data File Utility (DFU)	2-53		
Query Utility	2-54		
Text Management Program RPQ	2-56		
Remote Job Entry Facility (RJEF)	2-57		
Conversion Reformat Utility	2-57		

About This Manual

PURPOSE OF THIS MANUAL

The purpose of this publication is to help the reader who has a basic knowledge of data processing understand System/38 (a new general-purpose system) and its capabilities.

The manual summarizes what IBM System/38 is and how it can be used to meet an organization's data processing needs.

ORGANIZATION OF THIS MANUAL

This manual contains four chapters and a glossary.

- Chapter 1 summarizes System/38 design and highlights of its major functions.
- Chapter 2 gives a fundamental description of System/38 data processing concepts in terms of the System/38 program products and machine.
- Chapter 3 describes the system configurations and the machine characteristics.
- Chapter 4 describes the compatibility that exists between System/38 and other systems, and conversion aids that are available to help you move from other systems to System/38.

SUMMARY OF CHANGES

Information about the following release 4 items has been added to this edition of the manual:

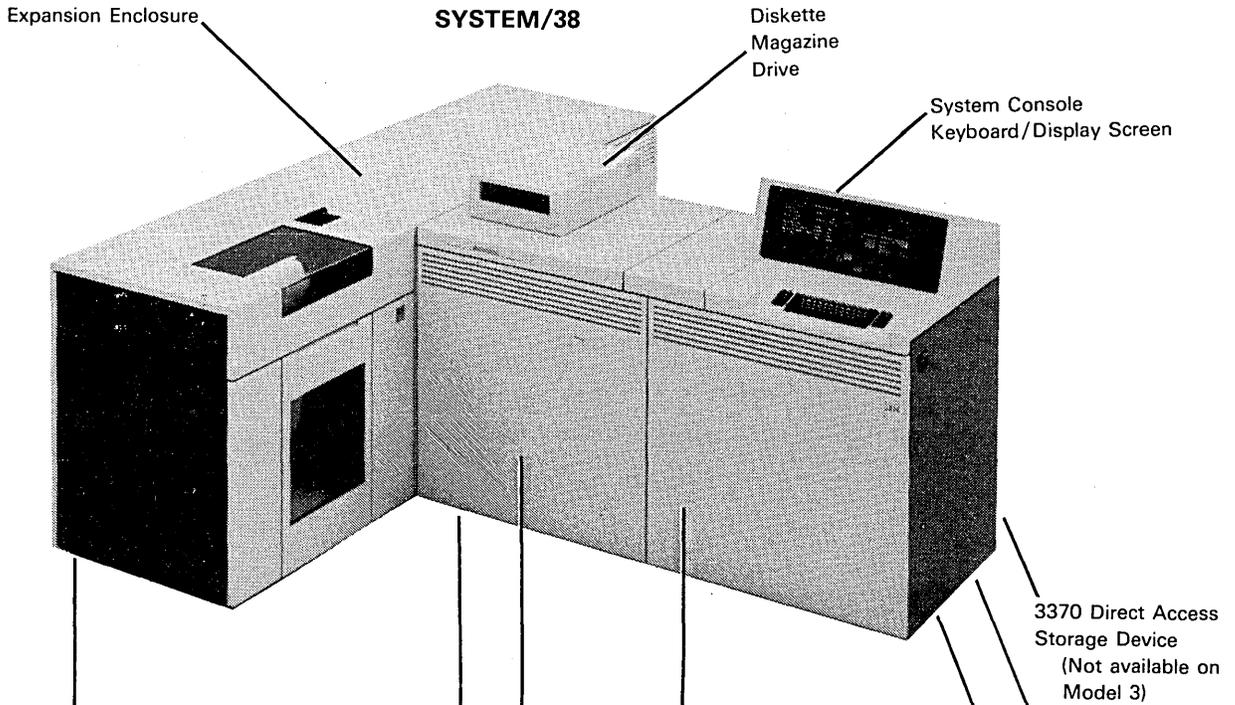
- High-speed local attachment of another System/38, a System/34, or a Series/1 to System/38
- IBM System/38 Model 7
- Text Management Program RPO
- Remote Job Entry Facility
- Auxiliary Storage Save/Restore Function

Changes are noted with a vertical change bar to the left of the changed section.

Note: This publication follows the convention that *he* means *he* or *she*.

If You Need More Information

The *IBM System/38 Guide to Publications*, GC21-7726, contains the titles and reading sequence of related publications.



Printer

- 5211 (Prints up to 300 lpm)
- 3262 (Prints up to 650 lpm)
- 3203 (Prints up to 1200 lpm)
(Up to two printers can be attached)
- 5225 (Prints up to 560 lpm)

Communications

- Work Station Controller (Local)
(Up to 80 work stations)
 - 5250 Information Display System
- Communications Attachment (Remote)
(Up to 8 lines)
 - Integrated Modems
 - DDSA
 - Auto Call (U.S.)
 - EIA/CCITT
 - SDLC Communications Control
 - 5250 Information Display System
Secondary to System/370
 - BSC Communications Control

Series/1	3741
System/3	3776
System/32	3777
System/34	3780
System/38	5110/5120
System/370	5230
30XX	5260
43XX	5280
CMC II	5520
OS/6	6240
	6580
	6640
	6670
 - High-Speed Local Attachment
 - System/38 to System/38
 - System/38 to System/34
 - System/38 to Series/1

Base Enclosure

- 5381 Processing Unit
(Model 3)
(Model 4)
(Model 5)
(Model 7)
- Main Storage
 - 512 – 1536 K (Model 3)
 - 512 – 2048 K (Models 4 and 5)
 - 2048 – 4096 K (Model 7)
- Auxiliary Storage
 - 64.5 – 129 MB (Base Enclosure)
 - 64.5 – 258 MB (Expansion Enclosure)
 - 571.3 – 2285.2 MB (1-4 3370s)
(Maximum auxiliary storage is 2672.2 MB)

Card Input/Output

- 5424 Multi-Function Card Unit
(Reads up to 500 cpm)
(Punches up to 120 cpm)
(Prints up to 120 cpm)

Magnetic Tape Input/Output

- 3410/3411 Magnetic Tape Subsystem
(Data rate of up to 80 000 bps)

Chapter 1. Introduction to IBM System/38

IBM System/38 is a general-purpose, data processing system. System design greatly reduces the effort usually needed to develop, maintain, and enhance applications for interactive work stations in an integrated data management system.

The system provides 512 K to 4096 K bytes of main storage, 64.5 to 2672.2 megabytes of nonremovable auxiliary (disk) storage, a diskette magazine drive, and a system console with keyboard and display. A multifunction, 96-column card reader/punch and up to two system printers, which operate at rates of up to 1200 lines per minute, can be attached. The system supports direct attachment of up to 80 local 5250 work stations (displays and printers) and, through teleprocessing lines, a large number of remotely attached work stations. Two communications attachments provide system control for the direct attachment of up to eight remote communications lines. Either BSC or SDLC line protocol can be operator-selected for use on each line. The 5250 devices require SDLC protocol.

You can place the work stations where they are needed in your organization (in the offices, departments, plants, and warehouses) so that your personnel can share a common data base and the processing power of a computer. Online, realtime work stations let your workers share programs, data files, and system resources without significant delays to any user. Also, System/38 lets them access data records by more than one key ordering path. For example, they can access inventory item records by item stock number while someone else is accessing these same records by vendor number.

System/38 support lets you maintain up-to-the-minute business data that you can make available to any work station user. Having information that is current, accurate, and quickly available to key personnel can greatly improve the efficiency of your organization.

On most current systems, the unique functions required by the work station applications either have been added onto the existing batch programming system or they are programmed by the user in the application programs. In contrast, System/38 was developed specifically for the work station environment. The function needed to support the work stations, such as data base management, has been designed into the system. Major advances in the level of system function, and the ease of use of that function, let you develop work station applications in less time than is needed for traditional systems.

System/38 supports a wide range of applications—from those consisting almost entirely of batch processing to those that make extensive use of work station applications. System/38 lets you make transitions through this range of environments while making optimum use of system resources. In general, installations that make full use of work stations still require the execution of a considerable amount of batch work. Therefore, System/38 manages a mixture of batch and interactive work, providing both fast response to work station user requests and good throughput for batch jobs.

To develop System/38, IBM first determined what functions were required. Then IBM determined which parts of these functions should be performed by the control program and which parts should be performed by the *machine*. Next we designed a machine with a functionally high-level interface (Figure 1-1). The machine does much more than just add, subtract, move data, and execute input/output operations. The machine performs complete functions, such as finding a file or program in system storage, locating a record in a file through a key, initiating a new process, manipulating queues, and allocating storage.

This high-level machine interface ensures compatibility between the Control Program Facility and programs implemented on every model of the machine. Also, new functions and performance improvements can be added to the machine without affecting the Control Program Facility or your application programs. New devices can be added to the system without reprogramming.

In summary, compared with traditional systems, System/38 provides you with an improved level of productivity, ease of use, data integrity, functional capability, and reliability.

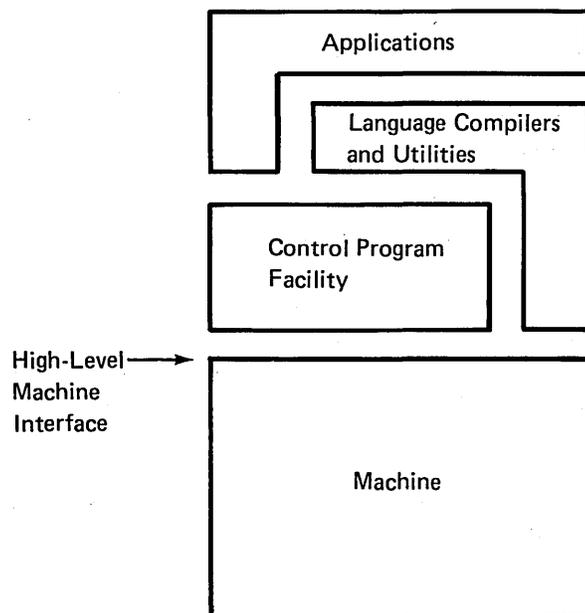


Figure 1-1. User and Machine Interfaces

PROGRAM PRODUCTS

The program products developed for System/38 include:

- Control Program Facility
- RPG III
- COBOL
- Interactive Data Base Utilities
 - Source Entry Utility
 - Data File Utility
 - Query Utility
 - Screen Design Aid Utility
- Text Management Program (RPQ)
- Remote Job Entry Facility
- IBM System/3 Batch Conversion Utilities
- IBM System/3 Communications Control Program (CCP) to System/38 Conversion Aid
- Conversion Reformat Utility
- IBM System/34 to System/38 Conversion Aid

Control Program Facility (CPF)

The Control Program Facility is the licensed system support program for System/38. Its many integrated functions help satisfy the total installation requirements for a multiprogrammed, batch, and online interactive system.

Some of the major new System/38 functions are:

Data base management that provides multiple, immediately maintained access paths. These paths can be shared concurrently by multiple users and by programs that perform read, update, add, and delete record operations. These functions are directly supported in RPG III and COBOL. The Control Program Facility also lets RPG III and COBOL process generic groups of records (for example, in an order entry application, access an order header record by key and then read the line item records sequentially). You do not need to sort or rearrange data for use in application programs.

Capability to design and code any application as if there will only be a single user of the application. All your application programs can be designed and coded as if there will be only one person using the program at a time; no special programming is required for multiple users. Theoretically, any number of work station users can execute any program at the same time. Although you should be aware of resource-sharing considerations, the system minimizes this application development requirement. You never need to specify the number of users. The concept just described is called the *multiple logical copy* concept, because even though there is only one copy of the program in main storage, that copy is used as if multiple copies were stored.

Work station data management. These functions let you define display screen formats outside the program and easily perform interactive functions with a minimal amount of coding. RPG III and COBOL use the Control Program Facility to take advantage of these functions. Data description specifications (DDS) define display screen formats to the Control Program Facility.

A control language. The control language (CL) is the set of all commands with which you request functions. CL consists of commands that have consistent, easy-to-learn names and syntax. Displayed prompts help you specify command parameters.

Control language commands can be used in two ways:

1. They can be used alone to request specific system functions.
2. They can be written in sequence and compiled into a control language program that tailors your system to the exact requirements of an application. Within CL programs, individual commands can be used to:
 - Control logic
 - Establish and modify variables
 - Manipulate data
 - Write data to work stations
 - Read data from work stations
 - Control arithmetic functions

Command entry interface. A command entry interface lets you enter control language commands, interactively, from any work station. If you need help while you use these commands, you can request command prompting, which provides formatted display screens that identify the parameters, their default values, and all the values that the system will accept.

A multi-user system. All system functions are available at all work stations through the control language. The system lets all work station users perform system functions concurrently as if each were the only system user. For example, several programmers can all be entering source programs and can be compiling programs at the same time. Also at the same time, batch and interactive applications can be executing. To each of these users, it appears as if he has the full use of the system for himself.

Externally described data. The programmer can describe each data field in data base records, in display records, and in printer records. On traditional systems, the data fields are described in the application programs; on System/38, through the external data description function, the field definitions are contained and maintained in the data base. This function eliminates the requirement to define the data fields in each program, provides more consistent definitions, reduces the data base maintenance effort, and reduces the amount of program change required when changes are made to the data base. (For conversion purposes, System/38 also supports program-described data as on traditional systems.)

Online program development functions. These functions let you easily:

- Enter and update source statements online
- Invoke the RPG III and COBOL compilers
- Compile control language programs
- View the results of compilation
- Protect data files while testing programs
- Debug the programs, using the debug function

Work management functions. These functions make it easier to control all forms of work on the system. This work can range from simple command execution to (1) batch job scheduling and execution or (2) interactive job control from work station sign-on to sign-off. Batch jobs may be submitted either from a conventional batch job stream or by a work station user. Once a work station user submits a batch job, he need not wait for its completion before continuing with other tasks. All batch work is queued for execution by the system, which schedules all work for throughput efficiency.

Spooling functions. System/38 spooling functions provide enhanced input and output operations. All job input from diskette, card, or the data base can be spooled (written to auxiliary storage) and queued for execution by the system. A great deal of flexibility in how the system selects these jobs for execution is provided to the user.

The spooling function (which sends the output of a job destined for the printer, diskette, or card punch to auxiliary storage) breaks the traditional job-execution limitation imposed by the availability or speed of the output devices. Spooling is especially important in the multiprogramming or multi-user environment where the number of jobs executing usually exceeds the number of available output devices. With output spooling, the output can easily be redirected from one device to another. For example, rather than being printed on a work station printer, the output can be redirected to the system printer.

Security functions. Because security functions are increasingly important in a work station environment, System/38 lets you protect data and other system resources so that unauthorized persons do not have access to them.

You can specify the level of security desired. The security level can range from a minimal level (in which you take all defaults and merely protect files from inadvertent destruction) to a more secure system designed to allow objects to be owned and accessed only by their owners.

The security function also lets one individual (such as the DP manager) define a personal *user profile* for each system user. This profile (1) uniquely identifies that user to the system, (2) requires the use of his own password, and (3) lets you give him access to certain files and programs.

Application development aids. System/38 lets you store (in the data base) a detailed description of each file, record, and field. Thereafter, you never need to describe the same item again. As necessary, you can print or display the descriptions as a current level of documentation. Another application aid, file reference, provides extensive information about the use of your programs, files, record formats, and fields.

Service functions. Your service representative can usually check input and output devices and diagnose system problems while you are using the system. This is an improvement over many traditional systems.

Message handling functions. System/38 messages are used extensively for:

- Error notification
- Logging activity
- Machine/operator communication
- Sending and receiving messages
 - To and from the system operator
 - Between work station users
 - To and from user message queues
 - To and from program message queues

System/38 first-level and second-level messages report errors reducing the need to look up error information in message manuals. The first level shows you a brief description of the error condition. If you want more information, you can request the second-level message, which further describes the error and gives corrective information.

Message handling lets you develop work station applications that require data queuing facilities. For example, the entry of an order requires examination for credit approval by someone in the credit department. The order (the order number) can be held in a message queue for processing by an interactive application that is executed by someone in the credit department. Additional items may be added to the queue from several work stations while existing items in the queue are being processed.

System/38 lets you store messages in message files. Then, each time any program must send a message, the program identifies which message is to be sent. The availability of all messages to all programs reduces the number of messages that must be written and stored in the system.

RPG III

RPG III includes many new language functions, such as:

Externally described data. Externally described data reduces coding and field definition effort, letting you add new applications quickly.

Full procedural file specification. This function lets you process the same file both randomly and sequentially in the same program. This file specification lets you easily develop many common applications.

File operation codes that do not use the RPG III cycle. RPG III file operation codes allow such functions as open, close, read, write, and update operations to the data base. Also, special work station input and output operations can be specified directly in the calculation logic. These operation codes provide a straightforward approach to interactive applications and an alternative for complex batch programs.

Separately compiled subprograms. System/38 lets any separately compiled RPG III program call another program and be called by other programs, along with the passing of parameters. These programs can be written either in RPG III, COBOL, or control language. There is no requirement to combine these programs in any manner (such as a link-edit operation) prior to execution.

New logic control operations. RPG III provides new logic control operations, such as various forms of the DO and the IF operations.

Data structures. RPG III lets you manipulate related data either as individual data fields or as a group.

COBOL

System/38 COBOL is designed according to the specifications of ANS COBOL X3.23-1974 and December 1975 Federal Information Processing Standard (FIPS Pub 21-1), low-intermediate. It contains significant ANS 1974 Level 2 function, extensions for data base and work station support, and enhanced programmer services.

Data base and work station processing extensions include:

- Externally defined data and display device descriptions
- Data base file extensions
- Subfile extensions
- Work station extensions
- Local and remote work station transparency

Programmer services include:

- Enhanced documentation
- Batch and interactive syntax checking

Interactive Data Base Utilities (IDU)

The IBM System/38 Interactive Data Base Utilities is a package of data handling utilities. They help you:

- Enter and maintain source statements for Control Language (CL), RPG III, COBOL, Data Description Specifications (DDS), and Reformat Utility Specifications
- Create and maintain application displays, menus, and menu CL programs
- Test application menus and displays
- Enter, change, delete, and verify data records
- Display data records
- Sequence, summarize, tabulate, and display or print information from data files

Source entry utility (SEU). SEU helps you enter and maintain source data. For example, source statements for programs (RPG III, COBOL, and control language) and data description specifications can be entered from a work station through SEU. SEU provides the following programmer aids: prompting, format control, automatic statement sequencing, syntax checking, and the ability to view several statements of a program while working with an individual statement.

Data file utility (DFU). DFU lets you quickly define data-entry, inquiry, and file maintenance applications without writing a program. On previous systems, if you wanted to define a data file utility application, you had to describe each of the data fields (such as item number, description, quantity on hand) to which data was to be entered, displayed, or changed. With the System/38 data file utility, this application definition is greatly simplified because the data fields to be used by that application have been previously described in the data base. Therefore, you merely name the file to be used. In addition, prompting support gives you several display formatting and error checking options.

Query utility. The query utility lets you interactively specify criteria for the extraction, summarization, and presentation of information from the data base. For example, you can easily generate a table presenting dollar sales and percent of total sales by product group and sales territory. Consequently, many applications that usually require considerable effort to program, using a high-level language such as COBOL or RPG III, can be quickly and easily specified with the use of the query utility. You can have the answer to your query displayed or printed or both.

Screen design aid (SDA). SDA lets you interactively design, create, maintain, and test display formats and menus that are to be incorporated into your applications. SDA generates data description specifications (DDS) for the displays and menus you design, and also generates the CL program source code needed to support your menus. Additionally, SDA compiles the menu CL programs and creates the display device files.

Text Management Program RPO

The Text Management Program lets any work station user create, modify, and store letters, forms, and similar documents. Text can be displayed on a split screen for editing, and the program can automatically print running heads or feet and can automatically number lines. Text management lets you perform the text processing function concurrently with other system functions.

Remote Job Entry Facility (RJE)

The Remote Job Entry Facility lets a System/38 function as a remote job entry (RJE) work station for submission of jobs to a central System/370, 30XX, or 43XX, using binary synchronous communications (BSC) over a point-to-point switched or nonswitched line. The remote job entry facility can be used for applications that collect data at your System/38 and send that data to the host system for processing. The host system can also send data to your System/38. For example, the host system can accept data from your System/38, process that data, and then send the processed data to your System/38.

System/3 Batch Conversion Utilities

The System/3 Batch Conversion Utilities for System/38 help System/3 users convert RPG II programs, operational control language, and data files to System/38. The use of these utilities shortens the time required to convert from one system to another. The System/3 Batch Conversion Utilities let the user do much of the conversion on his System/3. Therefore, he can accomplish the actual move to System/38 quickly and minimize the disruption of his data processing operation.

Communications Control Program (CCP) Conversion Aid

The CCP Conversion Aid is the group of System/3 programs and procedures that aid in converting System/3 CCP applications written in RPG and in using the display format facility (DDF) to describe 3270 display files to System/38 applications. This aid lets a System/3 user do a large part of his conversion on System/3 before his System/38 arrives.

The use of this aid removes many time-consuming tasks, leaving more time for you to concentrate on new applications to take advantage of the System/38 capabilities.

Conversion Reformat Utility

The Conversion Reformat Utility duplicates all functions of the System/3 Sort Utility, except summary sort and magnetic tape input and output. The System/38 Conversion Reformat Utility creates a record address file or a reformatted physical file. In addition, the Conversion Reformat Utility uses System/3 sort specifications. Although most sort requirements can be satisfied by the use of access paths on System/38, the Conversion Reformat Utility more closely duplicates the System/3 Sort Utility and gives you a quicker conversion from System/3.

System/34 to System/38 Conversion Aid Licensed Program

The System/34 to System/38 Conversion Aid helps you migrate from System/34 to System/38. Executing on a System/34 and a System/38, this aid converts System/34 RPG source code, operation control language (OCL), displays, sorts, and menus into code that can be processed on System/38. This aid automates many repetitive, time-consuming tasks, letting you devote more time to new application development.

MACHINE

System Unit

The IBM 5381 System Unit is the major unit in System/38. It is compact and has relatively low energy requirements for power and cooling. The 5381 contains:

- The processing unit and main storage for the system
- A system console with keyboard, operator/service panel, and display screen
- A diskette magazine drive
- Auxiliary disk storage
- A work station controller
- Space to house special features like the various I/O attachment features

The processing unit is the control center of the system. It fetches and stores data, performs the arithmetic and logical processing of data, executes instructions, and controls data flow between main storage and input/output devices.

The processing unit offers from 512 K to 4096 K bytes of main storage. The auxiliary disk storage holds between 64.5 megabytes (standard) and 387 megabytes. Separate fixed-disk storage, the 3370 Direct Access Storage Device (available on Models 4, 5, and 7 only) offers additional disk storage from 571 to 2285 megabytes (one to four units). The maximum auxiliary disk storage on System/38 is 2672 megabytes. The auxiliary disk storage uses IBM nonremovable, sealed, direct access storage assemblies. Advances in this type of storage improve price/performance levels, and improve efficiency and security because there are no disk packs or data modules to mount or remove.

The system unit has few mechanical parts, has error detection/correction in key areas, and has high-reliability electronics, so System/38 has few errors.

System/38 uses the same addressing method for all data, programs, and files, treating all auxiliary storage as simply an extension of main storage. Therefore, the Control Program Facility and your application programs can be designed and implemented without consideration of the available main storage. This type of virtual storage lets the Control Program Facility contain an extensive amount of function without requiring a corresponding large amount of main storage. Also, the Control Program Facility does not have to be generated or assembled upon installation. Storage management lets you plan your applications without related programs being in main storage at the same time.

Storage management does not partition main storage, nor is there any requirement that you code programs within specific sizes to provide better performance or to fit everything into certain size areas. All main storage is shared with the exception that the batch and the interactive jobs will normally be assigned separate logical pools of main storage to prevent the batch jobs from using main storage so heavily that the interactive jobs do not obtain a desired response time. You are allowed to reserve sufficient main storage for interactive jobs. The remaining main storage is then available for batch processing.

As data base files are expanded on System/38, additional increments of auxiliary storage are automatically obtained, as needed, from any available location on all the attached auxiliary storage.

The system console is used primarily for interactive communication between the system and the operator. The console display screen has 16 lines, with 64 characters per line. The keyboard contains standard alphameric keys for entering alphabetic and numeric data, cursor movement keys to position the cursor on the display screen, and command function keys for requesting specific program functions.

An operator/service panel on the console has indicator lights to show system conditions, switches to control system operations, and keys to power the system on and initiate the microprogram load process.

The diskette magazine drive lets you write quantities of data to removable magazines for backup, data interchange, and historical files. The diskette magazine drive holds up to two removable magazines and has three additional slots for individual diskettes. Each magazine can contain up to 10 large-capacity diskettes. The fully loaded diskette magazine drive offers a capacity of up to 24 megabytes (in 2D format) of removable data.

Input/Output Devices

System printing can be performed with the IBM 5211, the IBM 5225, the IBM 3262, and the IBM 3203 printers. They are floor-standing units that are quiet in operation and easy to install. One model of the 5211 printer is available with a speed up to 300 lines per minute. Four models of the 5225, with rates up to 560 lines per minute, are available. Two models of the 3262 printer are available with rates of up to 650 lines per minute. The 3203 printer operates at rates up to 1200 lines per minute.

For the 3262 and 5211 printers, many standard and special-purpose print belts are available, including a 64-character belt on the 3262, which is optimized to a statistical analysis of commercial printing requirements. This print belt usually provides better performance than a nonoptimized print belt and supplies all the special characters required for most users, thereby reducing the requirement for print belt changes. Many standard and special interchangeable train cartridges are available for the 3203 printer. The 5225 is a line matrix printer that offers a wide variety of character sets.

The IBM 5424 Multi-Function Card Unit (MFCU) is available for System/38 users who are growing from systems equipped with a 96-column card input/output device. As an input device, the 5424 can read cards from either of two hoppers and can stack cards in any of four stackers. As an output device, it can punch, print, and stack cards in any of four stackers. Two models are available. The faster model can read up to 500 cards per minute; it can punch and print up to 120 cards per minute.

The IBM 3410/3411 Magnetic Tape units are available for System/38 users who are growing from systems equipped with a 9-track magnetic tape input/output device or who need larger capacity storage backup. You can attach one 3411 Magnetic Tape Unit and Control and from one to three additional 3410 Magnetic Tape units.

The IBM 5250 Information Display System offers a family of products that can deliver the capabilities of System/38 to work stations throughout your organization. The 5250 system offers tabletop display stations and printers that were especially designed for online interactive applications.

You can install the display stations and printers in branch offices, departments, plants, warehouses, and other remote locations where there is work to be done. Wherever the devices are located, each remains independent in operation. Therefore, work station maintenance can be carried out without disrupting other areas of the system.

The display stations have 1920-character display screens. Three keyboard arrangements are available: a typewriter-like movable keyboard with a 10-key numeric pad, a data-entry keyboard, and a data-entry keyboard with a proof arrangement.

The tabletop printers offer speeds of 40, 80, or 120 characters per second, depending on the model. These are bidirectional matrix printers that also offer a maximum print line of 132 characters and print position spacing of 10 characters per inch.

The System/38 work station controllers allow direct attachment of up to 80 work stations within the same local environment as the system. There is no need for common-carrier-provided services to attach the 5250 devices locally.

Up to eight communication lines are available. System/38 can communicate, using synchronous data link control (SDLC) support, with work stations (IBM 5250) in remote locations. Through SDLC support, System/38 can communicate as a terminal system to IBM System/370. Through binary synchronous communications (BSC) support, System/38 can communicate with the following terminal systems and devices:

- IBM Series/1
- IBM System/3
- IBM System/32
- IBM System/34
- IBM System/38
- IBM System/370, 30XX, and 43XX
- IBM 3741 Data Station/Programmable Work Station
- IBM 3776 Data Communications Terminal
- IBM 3777 Data Communications Terminal
- IBM 3780 Data Communications Terminal
- IBM 5110/5120 Computing System
- IBM 5230 Data Collection System
- IBM 5260 Retail System
- IBM 5280 Distributed Data System
- IBM Office System/6
- IBM 5520 Administrative System
- IBM 6240 Communicating Magnetic Card Typewriter
- IBM 6580 Displaywriter System
- IBM 6640 Document Printer
- IBM 6670 Information Distributor
- CMCII Communicating Magnetic Card Typewriter II

Also, through BSC support, System/38 can communicate with the following systems in the local area at a nominal rate of 2400, 4800, 9600, or 56 000 bits per second:

- Another IBM System/38, by means of a cable connector that connects the cables from the DDSA (Digital Data Service Adapter) features in the two connected systems
- IBM System/34 , by means of a cable connector that connects the cables from the DDSA features in the two connected systems
- An IBM Series/1

Chapter 2. System/38 Program Products

The following IBM program products are available. They will help you efficiently operate System/38 and develop applications:

- Control Program Facility
- RPG III
- COBOL
- Interactive Data Base Utilities
- Text Management Program RPQ
- Remote Job Entry Facility
- Conversion Reformat Utility

These licensed programs will execute on all models of System/38. Therefore, you have the same *functional* capability whether you have the smallest system or the largest system.

CONTROL PROGRAM FACILITY (CPF)

The System/38 Control Program Facility is your primary interface to the system functions. It provides a much greater range of function than traditional control programs. Although you may not immediately need the full range of function provided by the Control Program Facility, after CPF has been installed the entire function is available when you need it. No conventional system generation is required; however, use of system specialization commands may be necessary to identify devices added to the system.

For ease of use, the most common options are assumed by default; that is, you do not need to specify them unless there is a special or unique requirement. For example, the system assumes that a printed report should be produced on the system printer on standard paper unless you specify that the report should be printed on a work station printer or on a special form.

IBM provides the necessary objects (those that are needed by a workable system). These include:

- A general-purpose library
- Source files for the most common types of source data
- Work station device and system operator message queues
- Subsystem descriptions
- Job queues and output queues
- Standard user profiles that are the default security definitions for the work station user, the programmer, the system operator, and the security officer

With the appropriate IBM education, the use of defaults and supplied definitions for system functions lets you install System/38 quickly and easily. Then, as you become more familiar with the many options available, you can select the specific options you want to improve your operation.

Figure 2-1 shows the major areas of function provided by the Control Program Facility. What is shown represents areas of functional support as you tend to view them, rather than the internal components of the Control Program Facility.

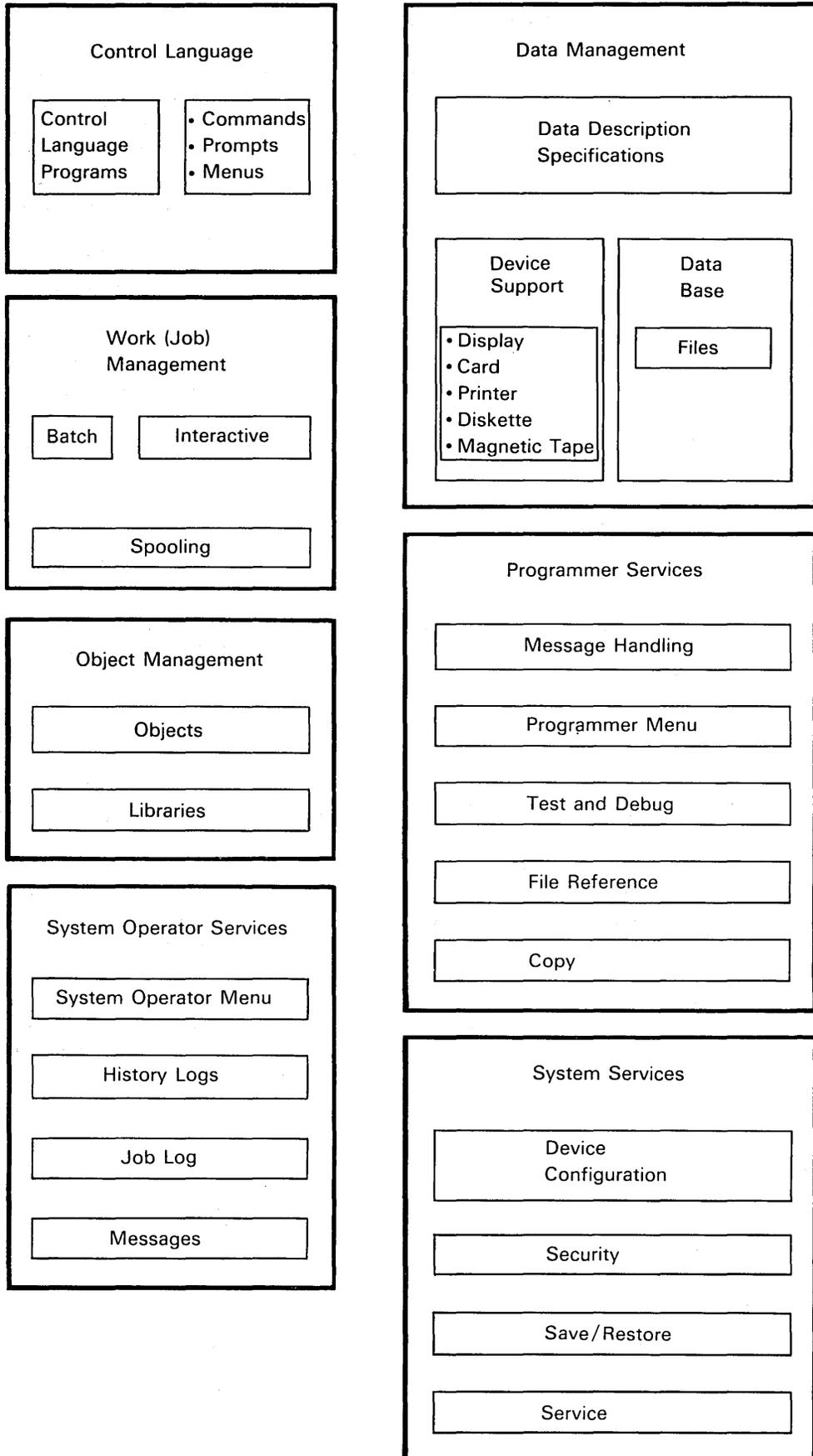


Figure 2-1. Control Program Facility Functions

User Interfaces

The Control Program Facility (CPF) gives you three user interfaces to the system functions:

- Control language
- Data description specifications
- Data management used by RPG III, COBOL, control language programs, and the Interactive Data Base Utilities

The system operator and the programmer commonly use these interfaces to control the system and to develop applications. Applications that are developed by the programmer provide the major interface (means of communication) with the work station user, such as an order entry clerk. Usually, the work station user is not expected to learn the control language and the data description specifications.

The work station user (such as an order entry clerk) might see the following typical series of displays:

```
Enter user password to sign on:
CAMERA (not displayed)
```

Control Program Facility (CPF) Supplied Display

The system identifies the user signing on as P JONES DEPT507 by associating the user's password, CAMERA, with his user profile. The user profile indicates he should initially be presented with the program call menu whenever he signs on.

```
PROGRAM CALL MENU
Select one of the following:
1. Call program (identify below)
2. Display messages
3. Send message to system operator
90. Sign off work station (*HOLIST *LIST)

Option: 1 Program name: ORDER
Parameters or message: _____
```

Control Program Facility (CPF) Supplied Display

The user selects option 1 (call program) and enters the name of the program, ORDER. (This display is optional.) The user profile could specify the name of a user program to be called.

```

                                ORDER DEPT GENERAL MENU
1. Inquire into Customer File
2. Inquire into Item File
3. Customer Name Search
4. Inquire into Orders for a Customer
5. Inquire into an Existing Order
6. Order Entry
7. End of Menu (RETURN to prior menu)

Option: 1

```

User Created Display
The program ORDER displays a menu of order entry functions. The user selects option 1 (inquire into Customer File).

```

                                CUSTOMER MASTER INQUIRY
Customer Number: 598702

```

User Created Display
A prompt requesting input of a customer number is presented. The user responds with 598702.

```

                                CUSTOMER MASTER INQUIRY
Customer Number 598702      Customer Type P
Name ACE Shoe Company
Address 523 Oak Street
City Chicago
State IL Zip Code 60611

Total A/R Balance 3,287.00
Credit Limit      10,000.00
Last Payment      1,529.13      Last Payment Date 06/19/78
Sales This Year   27,952.00
Sales Last Year   22,846.75

```

User Created Display
The customer information is displayed. When the user is finished viewing the display, he presses a function key to return to one of the previous displays.

A typical series of displays that the programmer might see are:

```

Enter password to sign on:
EARTH
      |
      | (not displayed)

```

Control Program Facility (CPF)
Supplied Display
The system identifies the programmer signing on by associating the password, EARTH, with the user profile for a programmer. This profile indicates he should be presented with the programmer menu whenever he signs on.

```

                                PROGRAMMER MENU
Select one of the following:
1. Design/execute DFU app      (app), ,(options)
2. Design/execute query app   (app), ,(options)
3. Create object              object name, type, pgm for CMD, (text)
4. Call program               program name
5. Execute command            command
6. Submit job                 (job name), (command)
7. Display submitted jobs
8. Edit source                (srcmbr), (type), (text)
9. Design display format      (srcmbr)
90. Sign off                  (*NOLIST *LIST)

Types:  BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 5_ Parm: _____ Type: ____ Parm 2: _____
Command: CRTLIB

Text: _____ Log requests: *YES
Src file: _____ Src lib: *LIBL Obj lib: _____ Jobd: QBATCH
CF3-Command entry CF4-Prompt (3 & 5 only) CF6-DSPMSG

```

*Control Program Facility (CPF)
Supplied Display*

The programmer menu lets the programmer enter control language commands. In this example, he wants to create a library. He knows the command is CRTLIB. If he could not remember the command name, he could request help from a command selection aid (menu). If he knows the command, but he cannot remember the parameters that go with the command, he can request *prompting* by pressing command function key 4.

```

                                Create Library (CRTLIB) Prompt
Enter the following:
Library name:                LIB      R      _____
Library type (*PROD *TEST):  TYPE    P      *PROD
Public authority              PUBAUT
(*NORMAL *ALL *NONE):       *NORMAL
Text 'description':          TEXT     *BLANK

```

*Control Program Facility (CPF)
Supplied Display*

The programmer is prompted to enter the library name, which is a required parameter, as indicated by the R. The other parameters, TYPE, PUBAUT, and TEXT are already filled in with their default values.

```

                                Create Library (CRTLIB) Prompt
Enter the following:
Library name:                LIB      R      ORDPROD
Library type (*PROD *TEST):  TYPE    P      *PROD
Public authority              PUBAUT
(*NORMAL *ALL *NONE):       *NORMAL
Text 'description':          TEXT     'ORDER PROCESSING LIBRARY - PR
DUCTION'

```

*Control Program Facility (CPF)
Supplied Display*

The programmer enters the name of the library to be created as ORDPROD. He also decides to enter text to describe the library. He then presses the Enter/Rec Adv key, causing the command to be executed. (If a parameter is in error, it will be displayed in reverse image with an error message at the bottom of the display.) After the system executes the command, it displays the programmer menu.

The system operator menu lets the system operator execute system commands.

```

                                SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)              8. DSPWTR (writer)
  3. SNDMSG tomsgq,(type),msg    9. SBMDTKJOB dev,label,(loc)
  4. CALL program                10. SBMDBJOB file,(member)
  5. Execute command            11. DSPSBMJOB
  6. SBMJOB (job),(jobd),(cmd)  90. SIGNOFF (*NOLIST *LIST)
Option: _____ Params: _____
Msg or cmd: _____

Log requests: *YES
CF3-Command entry CF6-DSPMSG QSYSOPR CF7-DSPSBS CF8-DSPSYS

```

Options 1 through 4 and 6 through 90 select commands that the system operator uses most frequently. The operator executes the command by entering its option number and any required parameters on the menu or by pressing one of the CF keys indicated on the bottom of the menu. (The command name is shown in uppercase following the option number. Each lowercase word following the command name represents a parameter for the command.)

The system operator can also execute less commonly used commands by selecting option 5 on the menu, then entering the command after *Msg or cmd* on the menu.

```

08/26/81 13:33:46          OUTPUT QUEUES
- QUEUE NAME LIBRARY      FILES WRITER      STATUS
- QDKT        QGPL         2
- QPRINT      QGPL         0      QSYSVRT
- QPRINTS     QGPL         0
- QPRINT2     QGPL         5
- QPUNCH      QGPL         2

1-DSPOUTQ    4-HLDOUTQ    6-RLSOUTQ

```

As an example of how the system operator can use the operator menu, say he enters option 2 to execute the DSPOUTQ command. The system presents the output queues display, which lists all existing output queues. From this display, the operator can select a particular output queue for display, hold a particular output queue, or release a particular output queue.

```

                MESSAGES IN QUEUE - QSYSOPR
Job PAY3.DEPT25.000358 completed on 11/17/80 at 08:32:27
What time is order entry application being started today
?: At 12:00
Log version QSRV80123A library QSYS full and may be saved
job WRITE.DAC.000475 started on 11/17/80 at 09:05:23
Device QCARD96 hopper check (C,R) ERR:OUOU-ERRC-TTTTTTTTTTT
?: _____

CF6 - Remove a message    CF7 - Display all    CF8 - Remove all

```

If the system operator presses CF6 while the system operator menu is being displayed, the system presents a display of the messages in the system operator message queue.

```

08/26/81  13:35:05  SUBSYSTEM JOBS - QCTL      +++
JOB NAME  USER      NBR   TYPE   STATUS
_ QCONSOLE QSYSOPR  001554 INTER ACTIVE

1-DSPJOB  2-Spl files  4-HLDJOB  6-RLSJOB  9-CNLJOB

```

Pressing CF7 while the system operator menu is being displayed results in a separate subsystem display that shows the job name, user name, and job number of each job in all currently active subsystems.

Control Language

The System/38 control language provides a simple, consistent syntax for invoking system-related functions. The instructions that make up this language are called commands. These commands can be:

- Entered individually from a work station
- Used as source statements in a control language program
- Used in a batch job stream

Each command contains a command name and a set of parameters. The command name identifies both the action to be performed and the object of that action.

For example, to send a break message to all the work stations, the following command is used:

<u>SNDBRKMSG</u>	<u>MSG('This system will be available until 5:30 PM')</u>
Command Name	Parameter

On a display work station, one way of supplying parameter values for a command is by responding to prompts that are provided by the Control Program Facility.

The prompt is made up of a set of fill-in-the-blank requests that guide you as you enter the parameter values. (The Create Library [CRTLIB] command prompt is an example of a prompt that a programmer might see.) This feature can be especially helpful when you know part of a command but do not remember certain parameters. When you request a prompt after part of a command has been entered correctly, the system displays any parameter values that have already been entered. Prompting is available for every command. The system automatically checks each command you enter for correct syntax as you enter the command. For example, if you make an error when you enter a command, the system displays an error message. If you need more information about the error, you request a display of a second-level message by pressing the Help key.

If you are not sure which command to enter, you can request a set of menus to help you select a command. Figure 2-2 shows the command grouping menu. This menu is presented in response to a request for prompting when a command name is not entered. When you select one of the command groups, such as 2, the system displays a list of commands for the selected area (for example, the object and library commands menu). When you enter a command name on this display, the system presents the prompt, just as if you had entered the command name from the command entry display and requested prompting for the parameters.

```
                                COMMAND GROUPING MENU
Select one of the following:
 1. All Commands
 2. Object and Library
 3. Data Base
 4. Device File
 5. Device Management
 6. Programming
 7. Program Debug
 8. Message Handling
 9. Utilities (If installed)
10. Input/Output Spooling
11. System and Job Control
12. Subsystem and Class

Option: 2 Partial command name: _____ CF1 - Return
```

```
                                Object and Library Commands Menu
Select one of the following:
Object
RNM OBJ - Rename Object
MOV OBJ - Move Object
ALCOBJ - Allocate Object
DLC OBJ - Deallocate Object
CHK OBJ - Check Object
DSPOBJD - Display Object Description

Library
CRTLIB - Create Library
DLTLIB - Delete Library
CLRLIB - Clear Library
DSPLIB - Display Library
RPLLIBL - Replace Library List
DSPLIBL - Display Library List

Command name: CRTLIB _____ CF1 - Return
```

Figure 2-2. Command Grouping Menu

Control Language Programs

The control language is much more than a collection of system control commands. It is actually a high-level programming language for system functions. You can use control language commands to write a program, and then you can compile that program, assigning a program name by which your program or another program can execute it. Compiling provides much better execution performance and allows more flexibility in the use of the language for:

- Logic control
- Branching
- Use of variables and arithmetic expressions
- Use of subprograms with parameter passing
- Some input/output operations, including input/output to work stations and message queues

Control language programs are powerful procedures that consist of groups of control statements. The following list of commands is an example of a small control language program:

```
Ⓐ—/* CREDIT CONTROL PROGRAM */
Ⓑ—PGM PARM(&ORDTOT &CUSTNO &ORDNO)
Ⓒ— [ DCL &ORDTOT *DEC (7 2) /* TOTAL AMOUNT OF ORDER */
    [ DCL &CUSTNO *DEC 6
    [ DCL &ORDNO *DEC 6
    [ DCL &CRDLMT *DEC (7 2) /* CREDIT LIMIT */
Ⓓ—CALL CRDCKPGM PARM(&CUSTNO &CRDLMT) /* CALL CREDIT CHECK PGM */
Ⓔ— [ IF (&ORDTOT *GT &CRDLMT) +
    [ CALL CRDAPRV PARM(&ORDNO) /* CALL CREDIT APPROVAL PGM */
    [ ELSE CALL ORDPROC PARM(&ORDNO) /* CREDIT IS OK. PROCEED */
Ⓕ—ENDPGM
```

- Ⓐ Comment
- Ⓑ Define start of a CL program
- Ⓒ Declare variables used in this CL program
- Ⓓ Call another CL program
- Ⓔ Logical expression
- Ⓕ Define end of CL program

Control language programs can call or be called by RPG III, COBOL, and other control language programs. Control language programs can send output to and receive input from the work station. A complete application on System/38 generally consists of a combination of control language and RPG III or COBOL programs.

Batch Input Streams

Control language commands are also used in batch input streams, which can be entered from the diskette magazine drive, a card device, or a data base file. For example:

```
// JOB INVUPT JOBPTY(2)
CPYF FROMFILE(INVMST) TOFILE(INVMST2)
CALL INV203
CALL INV204
// DATA
28607R 062778
59051I 062778
.
.
.
// ENDJOB
```

Object Management

Objects

An object is a named item that is made up of a set of attributes (that describe the object) and the data portion of the object. Control language commands perform functions such as create, change, delete, or display an object. Therefore, on System/38, the term *object* refers to more than just files because it can include items such as programs, messages queues, and user profiles. Briefly, the set of attributes that are assigned to each object include such things as:

- Name—The name assigned to the object when the object was created
- Owner name—The name of the system user who owns the object
- Creation date—The date on which the object was created
- Object type—An indication of the contents of the object

Libraries

Libraries are used to organize objects into groups. Most objects can be placed in libraries. A library can group many types of objects. There are several advantages in using libraries. For example, an application that uses a group of named objects in one library can be switched to use a different group of objects having the same names in another library. You can do this without making any changes to the application or the control language programs that name the objects used in the application. You can change only the name of the library to be accessed or the order of the libraries in the current library list.

Testing an application is an example of this use of libraries. An application can use a set of data files with test data and programs simply by placing the name of the library that contains these data files and programs in the front of a list of libraries that are normally used by the application.

Using libraries provides organization, similar to chapters in a book. You can put related objects into a library, while you put other objects in a second library. The objects in the second library are related to each other, but are not related to objects in any other library. Therefore, maintaining objects is simplified because the programmer does not have to deal with a list of all objects on the system as one large group.

Security can also be simplified. Through security, you can control access to objects in a library by controlling access to the library itself.

Any number of libraries can exist on the system. As Figure 2-3 illustrates, the system provides some libraries, while you create others. The system provides:

- A *system library* that contains IBM program product objects. Any authorized system user can use the objects in this library or place other objects in it.
- The *general-purpose library*. This is the default library where user-created programs and data are placed unless you specify another library. Any authorized user can use this general-purpose library.
- A *temporary library* that is automatically created for each job executed on the system. Objects in this library are known only to that job (not to any other jobs on the system) and only exist as long as the job exists. An interactive job starts when a user signs on a work station and lasts until that user signs off. During his use of the work station, the user has his own temporary library. Multiple work station users each have their own temporary library even if they are executing the same application.

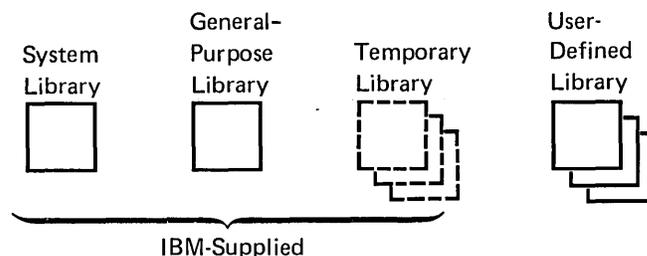


Figure 2-3. Libraries

In addition to maintaining the libraries that the system provides, you can define, rearrange, and delete any number of libraries.

Managing Work

A job is the basic unit of work on System/38. There are two types of jobs: interactive and batch.

An interactive job is a job in which the system performs processing actions in response to input provided by a work station user. During a job, a dialogue exists between you and the system. For example, between sign-on and sign-off you can create a physical file, delete a display file, and call a program. Each is a single command, but together they are a job.

A batch job is a job in which the processing actions are submitted as a predefined series of actions to be performed without a dialog between you and the system. The job consists of all the processing actions that result from input contained within the job. Basically, a batch job is all the commands between (1) a Job command and (2) an End Job command or the next Job command.

The work management functions allocate resources for use with jobs, manage jobs on the system, and schedule jobs for execution. The Control Program Facility provides all the functions necessary for a complete working system; however, you can use these functions where desired by issuing control language commands.

Some examples of the system's resources you can tailor are:

- Main storage
- Activity levels
- Input/output devices
- Job queues and output queues

Through the allocation of resources and through the use of priorities, work management lets you control execution of each job.

Subsystems

A job executes under the control of a *subsystem*. A subsystem provides a controlled environment for the execution of jobs.

The subsystem descriptions provided with the Control Program Facility supply generalized environments that meet the characteristics and requirements for the processing of interactive jobs, batch jobs, and input and output spooling operations. System/38 provides these subsystems:

- A *controlling* subsystem, which supports only the system console. The controlling subsystem starts automatically, when the system is started, to provide an interactive environment for communicating with the system.
- An *interactive* subsystem, which supports interactive processing through all the work stations attached to the system. The interactive subsystem supports work stations. All work station devices, excluding the system console, are attached to this subsystem.
- A *batch* subsystem, which supports batch job processing on the system. The batch subsystem provides a batch work environment. This subsystem can be started and terminated through the use of control language commands. Input to this subsystem comes from a job (input) queue. When the batch subsystem is not active, batch jobs can still be submitted for processing. These jobs are kept on the job queue until the batch subsystem is started.
- A *spooling* subsystem, which supports jobs that read batch job streams into the system from input devices and jobs that write output files to output devices. The spooling subsystem schedules and controls the system input and output through input/output devices (diskette, card, printer). Input can also be spooled from the data base.
- A *programmer* subsystem, which supports work done by the programmer. The programmer subsystem provides a separate environment to support the work done by a programmer. By isolating programmer work from other work, interference between the different types of work can be reduced and operational control can be eased.

In addition to the supplied subsystems, you can define your own subsystems. In this way, you can design work environments to execute specific types of applications. That is, if these generalized environments do not meet your requirements, you can use control language commands to tailor a subsystem to your needs or to create other subsystems.

For any subsystem, the system selects work for execution based on:

- Information you supply when you define the subsystem
- The system's capability to handle the additional work being considered
- Any special instructions supplied by the system operator

Spooling

System/38 has two forms of spooling: input spooling and output spooling.

- Input spooling is the function associated with job queues. Input spooling takes the information from the input device, prepares the job for scheduling, and places an entry in a job queue.
- Output spooling is the function associated with output queues. Output spooling lets programs write to an output queue rather than directly allocate the output device. Output spooling also offers flexibility in terms of which device can be used and when the output operation actually occurs.

Spooling can handle output that needs special forms, print belts, or printer forms alignment. Spooling can be used to print many copies and to stop and restart the printing of an output file.

With spooling, you can usually shorten job execution time and increase the number of jobs that can be executed sequentially while optimizing input and output device throughput.

System/38 spooling support for both input and output devices provides improvements in the following areas:

- Spooling usually improves throughput significantly. Because input and output devices are used efficiently, job execution is not bound by the availability of the device. For example, if an active program occasionally has printed output, a printer need not be allocated for the duration of that job, but only for the time actually required to print the output.
- Program design is much easier with spooling because you do not need to deliberately avoid device contention (that is, when two or more concurrently executing programs may require the use of the same device, such as a printer or card input/output device).
- Spooling simplifies application design for those jobs in which input from multiple work stations is destined for a single output device. The spooling function treats the output from different functions at different work stations as separate files.
- Because the program using spooling does not wait for slow-device completions, the program runs in less time and with better use of the system resources.
- A Copy Spool File command lets you place the contents of a spooled file into a data base file. The transferred data can then be used for further processing, can be transmitted to another system, or can be copied to tape for microfiche applications.

- System/38 spooling supports both local and remote devices. Each application program can accept input from, or direct output to, both local devices and remote work stations. The entire spooling operation can be controlled from any work station with little or no involvement by the system operator.
- Operation considerations are more easily handled with spooling. For example, rush jobs can be placed at the head of the input or output queue.

Throughput with Spooling

Figure 2-4 shows the throughput advantages you get with spooling. The upper part of the figure shows the processing of a System/38 job stream without the spooling facilities. Note that the first job, which needs a great deal of printing time, slows down throughput. The lower part of the figure shows the processing times of the same job stream with spooling. Here the total time is divided into processing unit time, queue time (the time the output of the job is in the print queue, ready for printing), and printing time. (Input spooling and multiprogramming are not considered in this example.)

When the system is spooling, the improvement in system throughput achieved by output queuing is shown by the difference in time between points 2 and 3. (Printing can begin prior to the completion of the program.)

Although the time elapsed between the reading of a job and the completion of its output increases when spooling is used, overall system throughput is usually improved. This can be seen from the difference between points 2 and 3, when all five jobs have finished processing. In addition, the processing unit is available for processing because between points 1 and 2 the only activity is the printing of the output queues, which requires little processing unit time.

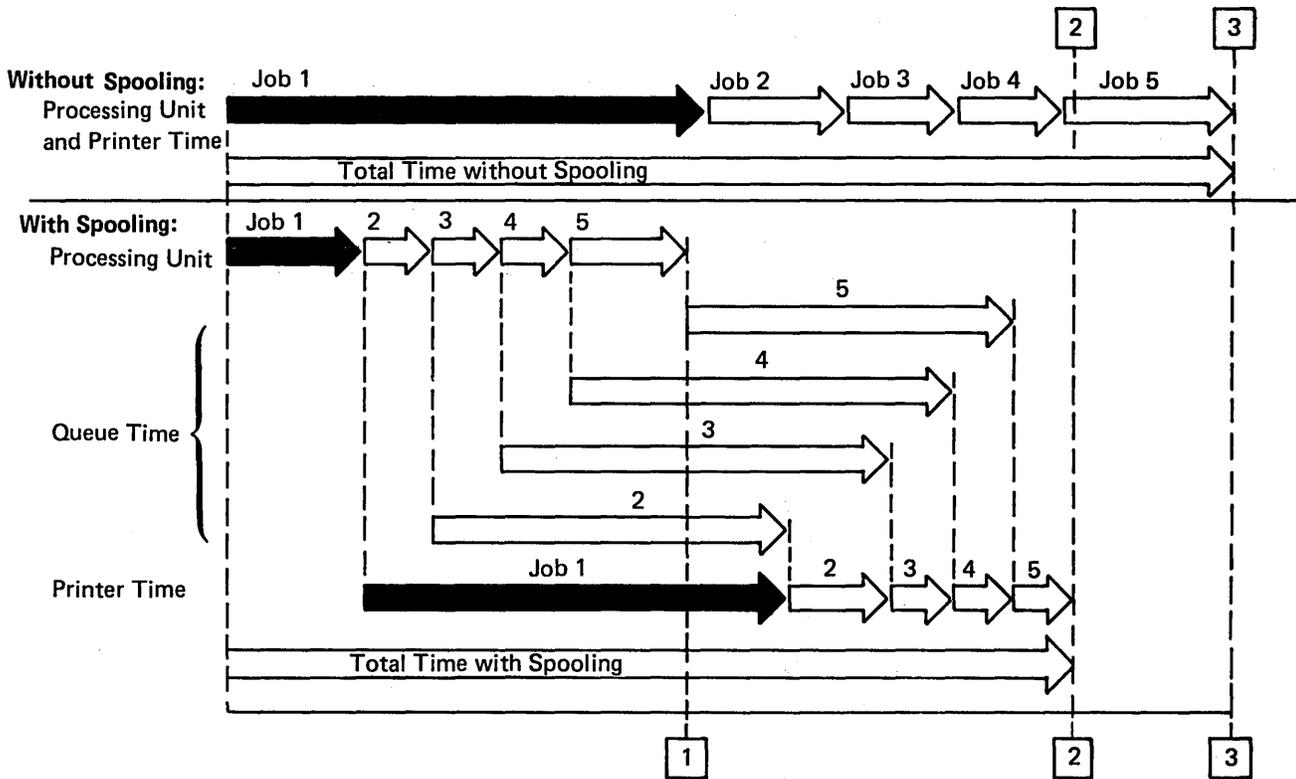


Figure 2-4. Processing Five Jobs with and without Spooling

Note that this example of spooling throughput in a single batch environment is theoretical. The actual increase in throughput depends on such factors as the I/O orientation of each program, the sequence of the particular jobs, and the number of unit-record devices and their operating rates.

Data Description Specifications (DDS)

On traditional systems, data fields for records on auxiliary storage are defined in the programs (for example, input specifications in RPG II programs).

On System/38, data description specifications let you describe the data fields external to the program (Figure 2-5). The description of each field (name, data type, length, decimal positions) and the location of that field within the record are defined through the data description specifications and are stored with the file in auxiliary storage.

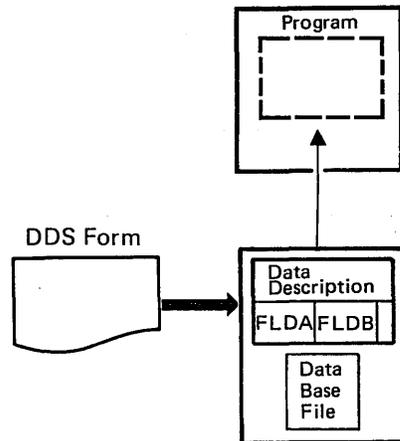


Figure 2-5. Externally Described Data

System/38 also uses data description specifications to describe record definitions for devices such as the work station. In addition to the usual field attributes (name, data type, and length), the work station data descriptions include many other important attributes, such as:

- Line and position attributes, on which the field should appear on the display
- Display attributes, for example:
 - Highlight (intensify or underscore)
 - Blink
 - Reverse image
- Field usage attribute (input, output, or both)
- Field editing attribute (insertion of decimal point) on output
- Field validity checking attributes (numeric only, mandatory entry, mandatory fill, allowable value range, allowable values by list, self-check modulus 10 or 11)

For System/38, you use the Data Description Specifications form (Figure 2-6) to describe data base records. The data description specifications are normally entered interactively from a work station by using the source entry utility, which provides prompting, format control, and syntax checking.

IBM International Business Machines Corporation		DATA DESCRIPTION SPECIFICATIONS										GX21-7754-0 UM/050* Printed in U.S.A.	
File EMPLOYEE MASTER		Keying Instruction	Graphic									Description EMPMA SP	Page of
Programmer		Date	Key										
Sequence Number	Form Type And/OR Comment (A/O/*) e Not	Conditioning					Name	Length	Reference (R)	Data Type (A/B/I/N/P/S/X/Y) Decimal Position Usage (U/D/E/H/M)	Location		Functions
		Indicator	Indicator	Indicator	Indicator	Indicator					Line	Pos	
1	A	*					EMPLOYEE MASTER RECORD FORMAT						
	A						R EMPMAS					TEXT('EMPLOYEE MASTER RECORD')	
	A						CODE	1				COLHDG('EMPLOYEE' 'RECORD' 'CODE')	
	A						CLOCK	4				COLHDG('CLOCK' 'NUMBER')	
	A						DEPT	3				COLHDG('DEPT.' 'NUMBER')	
	A						NAME	25				COLHDG('EMPLOYEE' 'NAME')	
	A												
	A												
	A						K CLOCK						
	A												
	A												
	A												
	A												
	A												
	A												
	A												
	A												

*Number of sheets per pad may vary slightly.

Figure 2-6. Data Description Specifications Form

1-10										11-20										21-30										31-40										41-50										51-60										61-70									
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6				
CLOCK										DEPT.										NAME										STREET										CITY																													
EMPLOYEE MASTER (EMPMA SP)																																																																					

Figure 2-7. Data Base Record Defined in Figure 2-6

Data description specifications are a new data specification language. They consist of a fixed-input format for the most common specifications (such as field name, field length, data type or a reference to the field reference file) and a free-form-keyword format for less-used or variable-length specifications. The data description specifications are used to describe display, printer, communications, BSC, and data base formats.

Several benefits result from separating data descriptions and programs.

- The high-level language programs (RPG III and COBOL) need less coding because the input and output specifications (data description) for data base files, display formats, and printer formats are not coded in the program. (The data description is required only once, in the file.)
- Greater control and consistency in naming and defining data is achieved.
- Application documentation is improved because text documenting the files, records, and fields can be stored with the data and displayed in compilation listings, cross-reference listings, and file listings.
- Column heading information can also be specified and is used to prompt for field information by the System/38 utilities.
- The definition phase of Interactive Data Base Utilities is simplified for data entry, inquiry, file maintenance, and query because the data is already described. Also, the system shows you the text descriptions of the fields to help you select those fields to be included in the definition.
- The file copy function is greatly enhanced because data descriptions of the from and to files are used to:
 - Rearrange fields in the record
 - Transform data types
 - Add or delete fields
 - Perform record selection

This capability replaces the need for many user-written programs.

- The owner of the file can secure specific sensitive fields while allowing access to, or the updating of, other less sensitive fields in a record.
- Record length and layout are no longer an important design consideration. You describe (including field length) each field in the record. The sequence in which you describe the fields determines their positions in the record. You need not specify the specific character position of the field in the record nor the record length. Different record types in the same logical file can be different lengths.

Data Base Data Management

Data base data management creates and manipulates online physical and logical data base files in auxiliary storage where the data and the file description are always stored.

You can describe files and record formats (using the data description specifications) outside the application program. This lets you use the same files in more than one program without having to describe the file each time it is used.

The System/38 data base capabilities include:

- Shared files—Multiple batch and interactive programs can simultaneously access, update, add to, and delete records in the same file. To provide integrity, System/38 locks a record (making that record unavailable to other jobs for update) when a user's job retrieves it for update. While the record is locked, it still can be retrieved by other users for read operations only.
- Described data—All data is known to the system. You can define unique physical files down to a field level. Each field description can contain a name, attributes, a text description, an edit code to be used when displaying the field, and validity checking specifications that can be applied when data for the field is entered from a work station.
- Multiple access paths—Many applications require access to a set of data (a data base file) by more than one key or index. For example, you might need to access inventory data by item number or by vendor number. In a work station environment, it is not practical to wait while the data is sorted into a second sequence. On some systems, you design a secondary index and write data management code in the application to access the data. System/38 supports multiple indexes to the data. The indexes on System/38 are called access paths. The enhanced functions and the immediate maintainability of the System/38 access paths provide significant new alternatives to application design and programming.

At your discretion, any or all of the multiple access paths that can be defined for a file can be immediately updated whenever any job makes a change to the file that affects the access path. For example, if a work station user accesses an inventory record by item number and changes the vendor number, another work station user accessing the same file by vendor number would immediately be able to access the record based upon the new vendor number (Figure 2-8).

You can apply selection criteria when defining an access path. For example, you may want to create an access path in customer number sequence within zip code, excluding all records with zero balances.

)
On System/38, an access path is the means by which the control program provides a logical organization to the data stored in the data base. When you create a data base file, you specify the access path for the file. When the file is used for input and output operations, records are retrieved according to the organization specified by the access paths.

You can access data, randomly or sequentially, by more than one access path (index). Each access path can define different fields as key fields and different ordering rules (ascending, descending, alternate collating sequence). At your discretion, each access path can be dynamically updated for any data change. In this way, different programs can process the records in the file in a different order without having to sort the records first. This capability eliminates the need to sort the data or to create and maintain complex indexes or chains of relative record numbers.

Data Base Files

Physical Files: A physical file is a data base file that actually contains data records. In this way, the System/38 physical file is similar to disk files on traditional systems. A physical file contains fixed-length records, all of which have the same format.

With physical file descriptions, you can specify a set of field descriptions that define a record format, which is stored in the data base. Data is stored in physical files in arrival order of records, but can be accessed in arrival sequence, keyed sequence, or by relative record number.

Changes to fields in the physical file do *not* have to impact any program if they are accessed through a logical file format, which automatically gets new transformation mapping to adjust the physical file change, if the fields in the logical files exist somewhere within the new physical file format.

Logical Files: A logical file is a data base file through which data from one or more physical files can be accessed in a format and organization that is different from the representation of the data in the physical files.

Any number of logical files can access the same physical file (Figure 2-8). Each of these logical files can define a different access path (index) for the data in the physical file and exclude and reorder the fields defined in the physical file.

A logical file differs from a physical file in that a logical file has no data in it, but contains a definition of how to retrieve and format fields from one or more physical files. The logical file contains a record format that can:

- Specify different field attributes than the fields in the physical record (data type, length, decimal position)
- Exclude fields contained in the physical record
- Rename fields contained in the physical record
- Create new fields by concatenating various fields from the physical record
- Reorder fields defined in the physical record
- Specify different key sequence/key fields, and select/omit logic to dynamically include or exclude physical file records mapped by the logical file

This capability, along with the ability to define multiple access paths (for example, record sequence), is essential in allowing each user in a multiple-user environment to view data in terms of his own application needs without requiring redundant copies of the same data. Any changes to a field within a record format require only the programs using that record format to be recompiled. Any changes to fields not included in the record format do not require recompilation. This significantly reduces the number of compilations required when data base changes occur.

For programs using logical files with their own record formats, you can change the physical file record format without recompiling programs (as long as the base fields remain in the same form). The data management function of the data base makes all necessary changes to provide the same logical record format to your program.

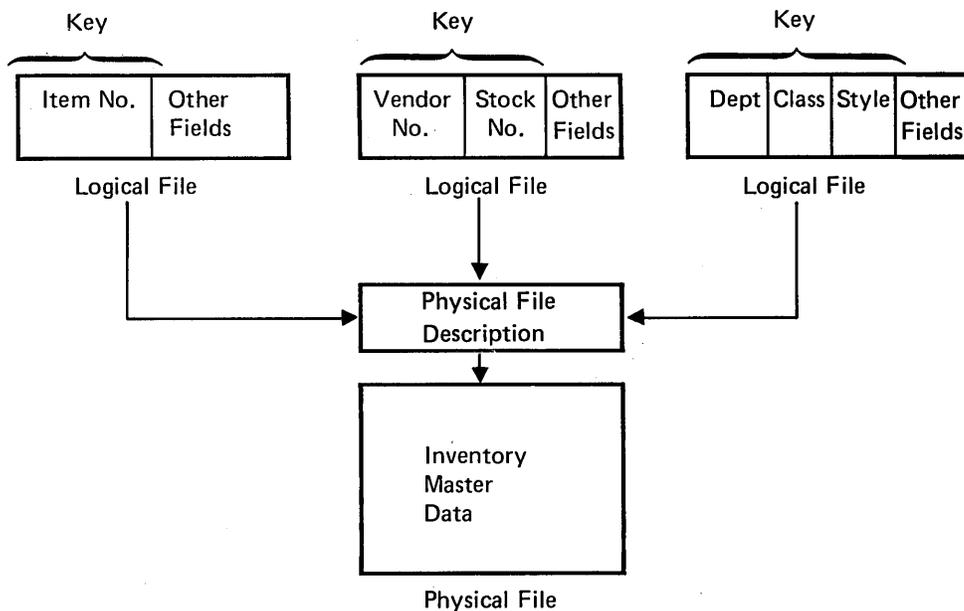


Figure 2-8. Multiple Logical Files Using a Single Physical File

Another benefit of logical files is that a logical file can be defined over two or more physical files (Figure 2-9), resulting in a logical file with multiple record types. The physical file record types need not be the same length. This results in a simpler file design than on many traditional systems in which each of the different record types in the file must be the same length, usually forcing you to assemble and disassemble (block and deblock) records in the programs.

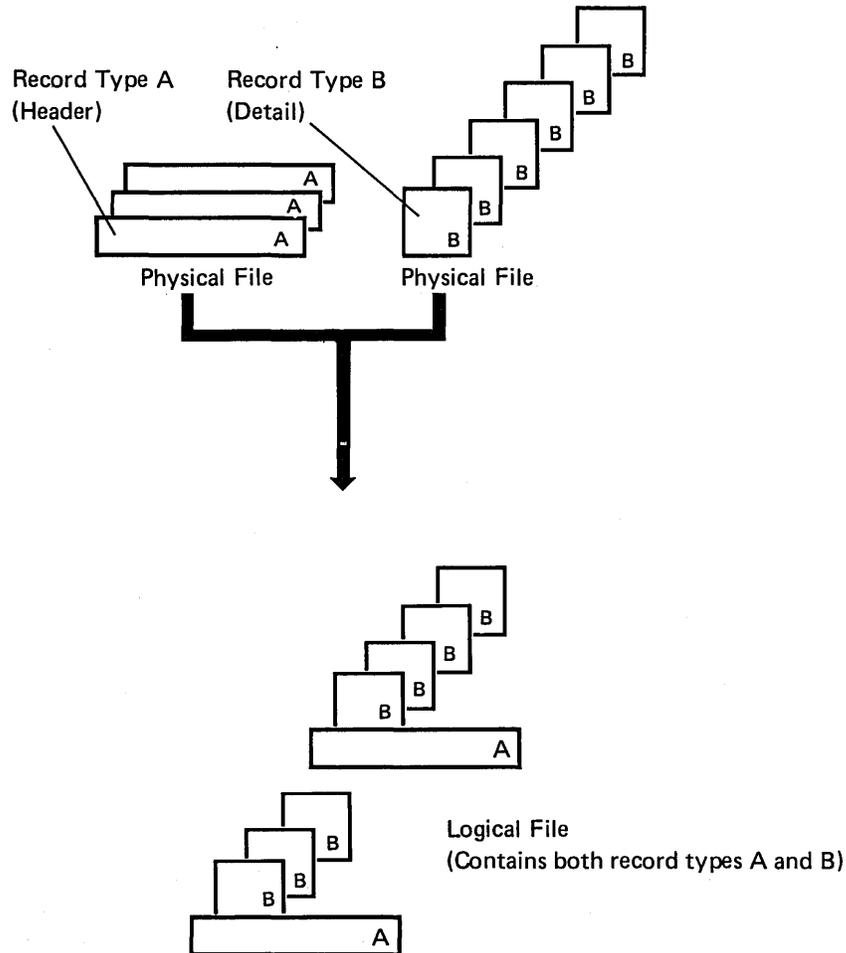


Figure 2-9. A Single Logical File from Multiple Physical Files

You can access data through both the logical files and the physical files. You can define as many logical file relationships as you want, whenever you want. The logical files look just like physical files to the programs; you do not need a special language to access them.

Data Base File Processing

System/38 provides two basic types of file organizations, which are determined by the access path: arrival sequence and keyed sequence.

In the arrival sequence file, all the records are arranged consecutively in their physical order (the order in which they are written to the file). Records in this type of file can be read, written, and updated using sequential or direct file operations. The direct operations use a relative record number to specify which record is to be read.

Keyed sequence file records each contain one or more fields that are specified as the *key*. The records are logically retrieved based upon the value of the key. Because a physical file can have several logical files defined from it, each of these logical files can specify a different key that defines a different order to the data records. Logical file records can be read, written, or updated using either sequential or random (by key) file operations. For sequential file operations, the next record in the keyed sequential order (determined by an access path) is normally read. However, the program can read the first, last, or previous record in the file. The program can also read the next record that is higher or lower than a specified key.

System/38 supports generic operations. This lets you process records with the same category keys easily. For example, an order header file record and its associated order detail file records, using the key *order number* can be grouped together automatically with a logical file. They can be retrieved in sequence automatically by *order number*.

System/38 data base data management provides many other file operations that simplify application development.

Data Base Logging

The data base logging component of the CPF (Control Program Facility) records all data base operations that alter the content of any data base file.

Under your control, data base operations are recorded within a routing step. You control, via commands, who logs changes, the files for which changes are logged, when changes are logged, and the data base file in which the changes are recorded.

The data base file in which the changes are recorded is called the log file, which you create. The system enters the changes by inserting records (log entries) into a specific member (log member) of the log file.

The functions that make up the data base logging component fall into the following categories:

- **Logging control:** Control of when logging begins and ends, specification of the files for which logging will be active, and specification of the log file and log member. The logging control operations include the Log Data Base Files command, the End Logging command, and routing step termination.
- **File level operations:** Operations on a file/member level that will be logged. These include open, close, clear physical file member, initialize physical file member, reorganize physical file member, and force end of data.
- **Record level operations:** Operations on a record level that will be logged. These include put, update, and delete operations.

Device Data Management

Device data management uses a device file description (stored in the system) to identify the characteristics of each type of System/38 input/output device. For each type of device, there is a device file that describes the device and the format of data for that device. These device files do not contain any data, but are used to build record formats and to control the transfer of data between the physical devices and the programs that use or generate the data.

Device data management support can be classified as follows:

- Support for display devices (for example, work stations and the system console), which generally use externally described data.
- Support for nondisplay devices (for example, diskette, card, magnetic tape, and printer devices), which generally use program-described data. (The printer can also use externally described data.)

Display Device Data Management

Display devices, such as the work stations and the system console, need more programming than card readers, printers, or a data base file. For example, for a field on a display device, in addition to such normal information as field names, length, data type, and decimal positions, you must specify:

- Position of the field on the display
- Type of field (input, output, or both)

You can also specify:

- Additional field attributes (such as highlight, underline, blink, reverse image, protected)
- Output editing (such as decimal point, comma)
- Input validity checking (such as numeric only, mandatory entry, mandatory fill, self-check digit, range check, list of values)
- Command function key and other special key specifications
- Displaying error messages
- Output and input of multiple repeated lines of data
- Various input/output operations to the device
- Many other functions, such as duplicating fields, overlaying information on the display, erasing portions of the display, positioning the cursor, and locking and unlocking the keyboard
- Print key support to let you write the contents of the display to a named device file or to a work station printer

All these functions are controlled through display device data management, in a way that reduces the amount of coding required to write work station applications. Most of this information is specified in external data description specifications, which are stored in a display device file (Figure 2-10). For example, an input field may have validity checking that specifies the field is mandatory entry, with a value from 1 through 5. If no data is entered or if anything other than a number from 1 through 5 is entered, device data management locks the keyboard, displays the field with reverse image, positions the cursor to this field, and displays an error message. All this is accomplished without any user application programming.

Your application program communicates with the display device file by using conditioning and response indicators (specified in the data description specifications) that can be set and read in the program. For example, setting on conditioning indicator O2 in the program can cause several optional fields to be included with the output to the display. Also, the entry of data in a specified field can cause response indicator O5 to be set on when the program reads from the display.

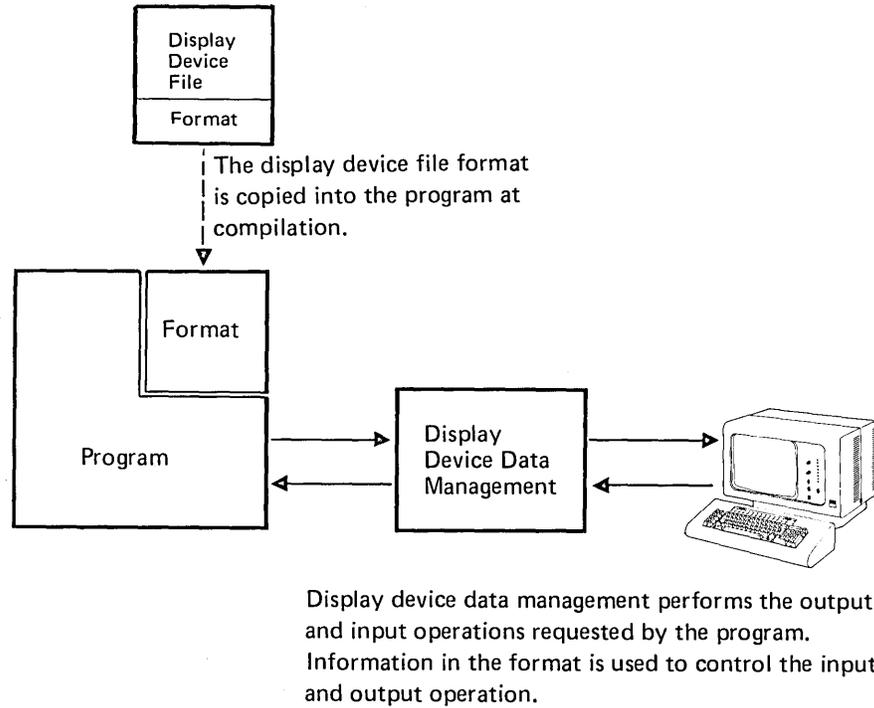


Figure 2-10. Display Device Data Management

The Control Program Facility performs all the operations needed to control the work station. The Control Program Facility formats data on the display screen, handles conditions that occur at the station, and controls the input and output operations at the station. In addition, this system support lets you create application programs that need not specify which work station is involved in the input and output operations.

Also, System/38 provides a subfile function. For this function, you identify a group of similar records from a data base file (for example, a page of line items on a customer order, the transactions against a checking account, or the charges on a customer's bill) as a subfile of records to be processed. Before the records are processed, they can be displayed on the screen as a long list of records that can be rolled up and down. The work station operator can make changes on the records as they are displayed; the system identifies every record that is changed. After any necessary changes have been made to the records, the system processes the data in the subfile. If requested to, the system will search the subfile for changed records, processing only those that have been changed. When all necessary processing has been performed on the subfile, the system deletes the identification from the changed records and returns the updated subfile records to the data base file. As you can see, the subfile function can greatly reduce the effort needed to process many applications.

Device file descriptions let you produce display screen definitions that can be used directly by RPG III or COBOL programs. You define the display screen once and do not need to redefine that display in the input or output specifications of the application program.

Diskette, Card, Magnetic Tape, and Printer Device Data Management

With diskette, card, magnetic tape, or printer files, the device file generally describes only the file characteristics. The data fields are described in your application program. And the system usually lets you switch program input and output from one device type to another. For example, output normally written to the diskette can be printed. All special file characteristics can be specified in the device file. In this example, a printer device file would specify page width, number of lines per page, and whether forms alignment is required.

Programmer Services

System/38 provides many aids to assist in online program development from either local or remote work stations. These aids let you:

- Use the programmer menu to simplify requesting programmer functions
- Enter source data online (using the source entry utility)
- Submit a job for compilation from the work station
- Display the output of a compilation at the work station, or print it on the system printer or on the work station printer
- Test and debug through CPF facilities during program execution
 - Breakpoint
 - Display variables
 - Dump
- Test libraries
- File reference facilities
 - Display file description
 - Display file field description
 - Program references
 - Display data base relations
- Move data between device files or data base files

Programs can also be entered by using traditional batch methods.

Testing and Debugging

Many System/38 functions are useful for testing application programs. These include the copy function, to copy portions of production files for use in testing; the data file utility, for creating and displaying files during testing; and the display spooling output file function, to allow viewing batch output without waiting for the actual printed output.

In addition, System/38 provides two special testing and debugging functions:

- Test library/test environment
- Debug function

When testing an application, you can operate in a test environment. When in the test environment, your programs can only update files in a test library; therefore, all files in the production libraries are protected from alteration. By using a library list in which the name of a test library is placed ahead of a production library, you can test programs and files from a test library that has names identical to the programs and files in the production library.

Testing can be performed concurrently with normal production.

The test environment is also useful for work station user training. In this situation, you create a test library containing files used for training. The worker being trained is free to alter any data in this test library, but does not alter data in the production libraries.

The debug function, which is usually used in conjunction with the test environment, lets you specify breakpoints in control language, COBOL, or RPG III programs. Breakpoints are points in a program at which execution is suspended to let you display and alter variables and indicators. You can specify breakpoints, variables, and indicators by control language commands that identify the source statement or variable in the control language, RPG III, or COBOL program (for example, the statement number or the variable name).

The debug function is used at execution time. It lets you examine the operation of a program in complete detail. You specify nothing when you compile the program. While a program is suspended at a breakpoint, you can enter control language commands to perform some other function. For example, you might stop a program right after the point at which it updated a file. Then you could use the data file utility to display a record in the file, to see if the update was performed properly, before you resumed execution.

Interactive debugging usually is used to debug both batch and work station programs. Also, a trace function provides a listing of the execution sequence of a program.

File Reference

System/38 provides a system-wide data base and device file reference function. You can use this function to get several types of cross-reference information. The file reference function contains information about:

- Data base definition
- Data base organization
- Data base usage
- Device file definition
- Device file usage

System/38 gets this information directly from the file descriptions, rather than from the source used to create the descriptions. Therefore, the information always represents what actually exists on the system.

This file reference information helps you:

- Design new applications that must interface with existing applications and files
- Change existing applications
- Find errors in existing applications
- Document applications

Besides detailed information about files and record formats, the following information is available:

- A list of all files that use a specific record format
- A list of all logical files that use fields from a specific physical file
- A list of all files referenced in a program (or group of programs)

Through optional command parameters, you can specify that you want file reference information to go to a data base file for analysis. Then the specified data base file will contain detailed information about each data field in a file or group of files. This information includes the organization of the data base, the field descriptions, and program references (data base and device files). Normally, this data base file would contain a very large amount of detailed information. Therefore, you would use a logical file over this file or use the query utility to select and display information from this file as that information is needed.

The file reference information can be used to answer such questions as, "What are the names of all the files that use FIELDX?" This information is useful if the definition of FIELDX is to be changed.

Message Handling

The system stores all messages generated by the Control Program Facility and other IBM program products. It stores the messages as data in separate message files that can be manipulated externally to the function that requests them. In this way, groups of messages can be tailored to meet the needs of various types of users (such as different experience levels and different national languages).

When sent, messages are received by message queues provided by the system. The message queue then delivers the message to the work station or job associated with the queue. You can change the message queue delivery operation to:

- Deliver a message immediately
- Notify the user that a message is waiting
- Hold the message for later delivery
- Reject the message
- Use the default reply

System/38 provides a permanent message queue for each work station, the system console, the system operator, and the history log file. In addition, the system lets you:

- Create message queues and message files
- Add, remove, change, and retrieve messages from the message file
- Route messages to various locations in the system
- Queue the delivery of messages at various locations
- Send messages to, and receive messages from, programs and work stations

Copy

You can use the copy function to move data between device files, data base files, or device and data base files. Unlike traditional copy utilities, the System/38 copy function is field-level sensitive. Therefore, if you do use the copy function, you can rearrange, enlarge, or drop any of the fields. The system also provides a way to define data base files.

The copy function, RPG III, and COBOL support data interchange between System/38 and other IBM systems using the diskette or magnetic tape device.

System Operator Services

System/38 is designed as a work station system that lets work station users perform work with less need for a system operator.

The system operator receives messages:

- That were sent to him explicitly by an application program
- That respond to work station user requests
- That relate to devices of the system unit

The system console performs much like any other work station, so the system operator can request various system-supplied or user-written functions. The system provides a system operator menu so that you can request common system operator functions easily (Figure 2-11).

```

                                SYSTEM OPERATOR MENU
Select one of the following:
 1. DSPJOBQ (jobq)                7. STRPRTWTR device,outq
 2. DSPOUTQ (outq)              8. DSPWTR (writer)
 3. SNDMSG tomsgq,(type),msg    9. SBMDTKJOB dev,label,(loc)
 4. CALL program                10. SBMDBJOB file,(member)
 5. Execute command            11. DSPSBMJOB
 6. SBMJOB (job),(jobd),(cmd)  90. SIGNOFF (*NOLIST *LIST)
Option: ___ Parms: _____
Msg or cmd: _____

Log requests: *YES
CF3-Command entry CF6-DSPMSG QSYSOPR CF7-DSPSBS CF8-DSPSYS
```

Figure 2-11. System Operator Menu

Because control language can be entered at any work station (local or remote), most of the system operator functions can be performed at any work station. In addition, the system operator menu can be used at any work station and the system operator's message queue can be switched to any work station so that the complete system operator function can likewise be performed at any work station. The only requirement is that the system console be used to power on the system and to respond to the initial startup display presented by the Control Program Facility. An option exists that permits the system to be started even if the system console is inoperable. In this case, one of the work stations can be used by the system operator.

Display commands let you display objects, entries in libraries, and status information. For example, the spooling function lets the system operator display input and output queues, as well as the actual data in an output spooling file.

History Log

The Control Program Facility maintains a system history log that contains all messages sent to the system operator. Each message is time-stamped. The history log supplies an audit trail of the status and state of the system, subsystems, jobs, and devices. This log can be printed or displayed by using a CL command. Programs can send messages to the history log. Also, the history log can be read by an application program.

A message is sent to the history log as every job begins, and a message is sent when every job ends. The job end message contains the CPU time used (in seconds) to execute the job, and a completion code that shows whether job execution was successful or unsuccessful.

Job Log

The job log gives you a record of the job requests entered and the results of those requests. If a job should terminate prematurely, the job log will contain the error information.

The Control Program Facility creates a job log at the start of each job. For batch jobs, the system creates the log when the job is read in. For interactive jobs, the system creates the log during sign-on at the work station. The job log can be printed with the job output for batch jobs or interactive jobs.

Messages

The Control Program Facility message-handling function processes both automatic responses and operator responses to messages. Operator-initiated messages can be sent from any work station to any other work station, including the system console.

Predefined messages can have two levels of text. The first level is the text which is initially displayed. The second level is additional text to be displayed if requested. The system operator and the work station user can request the second-level text by pressing the Help key.

System Services

The System/38 Control Program Facility includes functions that help you manage system operation. These functions include:

- Device configuration
- Security
- Save/restore
- Service (concurrent maintenance)

Device Configuration

In many systems a complex system generation function tailors the operating system to the desired support. The System/38 Control Program Facility is operational when it is installed and after the system devices have been defined. You can modify various defaults and create specific functions for applications to tailor the system further.

The primary requirement for the installation of the Control Program Facility is the entry of device configuration commands, which specify optional devices attached to the system. For example, the diskette magazine drive is standard, so it needs no configuration commands entered. Work stations and work station printers require entry of device configuration (device description) commands. Main and auxiliary storage are not considered to be devices, so no configuration command is required when adding storage.

Security

System/38 has two major security functions:

1. The integrity security function reduces the possibility of inadvertent destruction or alteration of programs and data by a work station user or someone not trained in data processing.
2. The user-authorization checking security function lets you restrict, to specific users, information that is accessed and displayed. This function prevents unauthorized disclosure and alteration of sensitive data.

Although IBM provides these System/38 security functions, you are responsible for using them correctly to make your system secure.

Both of these functions are especially important in the work station environment in which physical security is difficult to maintain. In the work station environment, there may be many untrained persons who can gain physical access to the data processing system. In many instances, it will be difficult, if not impossible or impractical, to completely control physical access to all work stations connected to the system.

In a work station environment, the entire data base is usually maintained online, and the minute-by-minute operations of the company become quite dependent on the proper functioning of the system. Therefore, integrity becomes very important. It is not acceptable for some data file to be inadvertently updated with the wrong set of data, thereby requiring a recovery process while everyone waits to go online again.

As a work-station user is identified to it, the system automatically determines which functions and application programs to call for that user. For example, at sign-on, the system might present an order entry clerk with choices for performing order entry applications, a credit manager with credit approval requests, and a programmer with the command entry display.

The System/38 security functions do more than simply limit access to the system through some mechanism such as a password. For example, each System/38 object has a specified owner (generally, the programmer). The owner of an object specifies who may use the object and how they may use it. The object owner must ensure that proper authorizations are initiated when the object is created, although authorizations can be initiated, changed, or cancelled later.

Once a user has signed on to the system by entering his user password, the system associates this user with his previously created user profile, which is stored in the system. The contents of his user profile determine which objects (such as files and programs) this user is permitted to access. In other words, on System/38, users are authorized to objects. For example, user RSMITH can be given access to the OPNORD file while user BJONES is not given access. Additionally, RSMITH could be given read and update authority while BJONES could have the authority to read only. This type of control (enforced by the machine itself) gives each user authorization only for the functions necessary to perform his job.

The security function is controlled by the security officer, a person operating with a special user profile called the security officer profile. Only the security officer (1) can create user profiles, (2) can display and alter user passwords, and (3) has full authorization to access any object in the system.

The security officer should understand all aspects of the system concerning commands, parameters, and defaults that deal with security functions so that he can administer system security effectively.

You can implement the security function easily by using the system-supplied user profiles and defaults. Then, as your company requires a higher level of security, you can use the more comprehensive security features.

Note: Even though a proper level of security can be established and maintained on System/38, that security level can be breached by:

- A restore operation that reads unsecure objects into your system from another System/38
- The use of customer engineer (CE) diagnostic programs that can display and alter secure information

Therefore, you should ensure that your console and diskette reader are used properly and only as you intended to have them used.

Save/Restore

System/38 machine and licensed programs provide maximum system availability and minimum user concern for recovery. However, with the fixed disk architecture and large disk file capacity of System/38, you should implement a thorough backup procedure to protect your system against possible machine, program, operational, and power failures.

The system provides two different ways to save and restore the system:

- The save/restore function
- The auxiliary storage save/restore (ASR) function

The save/restore function:

- Saves the entire system by writing to the diskette magazine drive
- Saves objects, groups of objects, and libraries by writing them to the diskette magazine drive or to the magnetic tape device
- Restores objects, groups of objects, and libraries that have been saved offline
- Optionally, makes available for use space that is currently holding data that has already been written offline. The system still keeps the object description and the offline location of that data.
- Restores only the most recently saved copy (The system operator can override this protection.)
- Displays save/restore status by volume

The auxiliary storage save/restore (ASR) function:

- Saves the entire system, including system libraries, user libraries, and access paths, by writing them to diskettes or tape.
- Restores the entire system, including system libraries, user libraries, and access paths that have been saved offline.
- Uses compression as it saves the system; this results in less time needed for I/O operations and less storage media.
- Writes data from storage to tape or diskette track by track, disk by disk without regard for file organization. The data is written back to its correct storage location during the restore operation (track by track and disk by disk).

ASR is not intended as a replacement for normal backup and recovery procedures that use CPF functions such as save/restore. You should continue to use these procedures to save and restore objects and libraries. The advantage of ASR is that it is faster than a library-by-library restore, and the IMPL recovery/rebuild operation is significantly reduced when ASR is used.

Service

The service representative can perform concurrent maintenance on many of the input/output devices on System/38. With the concurrent maintenance support, the system operator and the service representative can share the use of the system console. Information displayed at the system console can help the service representative diagnose problems faster.

The following items are available to help the service representative:

- History file (system operator messages, responses, and the status of the system at the time of failure)
- Error recording area (a defined storage area for accumulating machine and program errors)
- System level files (a defined storage area to hold current release and program change level information)
- Traces (levels are included for the type of function to be traced)
- Display facilities (allow a dump to the printer or the console)
- Modification facilities (allow permanent or temporary changes to the IBM system support)

RPG III

The RPG language is designed specifically for commercial applications. The fixed format of the specifications makes the language easy to learn, yet RPG III also offers the experienced programmer many advanced functions.

System/38 provides extensions to the basic RPG language that enhance its usefulness by providing the flexibility needed in a data base and work station environment.

Language Enhancements

System/38 RPG III offers many enhancements over previous RPG products.

- *Externally described data* lets you share data descriptions to minimize the input and output specifications for externally described data files. This provides ease of coding, consistent naming conventions, and improved growth capability. Many System/38 programs can be written with few or no input and output specifications.
- *Full procedural file specification* indicates the file is processed by programmer-specified calculation operations. A primary file specification is not mandatory. This lets you use many new application approaches in a straightforward manner.
- *Explicit input/output calculations* provide enhancements to the following:
 - Existing file processing operation codes
 - a. READ (read a specific record type or format)
 - b. CHAIN (chain to a specific record type or format)
 - c. SETLL (get an indication of an equal condition)
 - New file processing operation codes
 - a. READE (read equal for generic group processing)
 - b. READP (read previous for reading backwards along an access path)
 - c. WRITE (write a new record without the use of output specifications)
 - d. UPDAT (update of an existing record without the use of output specifications)
 - e. DELET (delete an existing record)
 - f. SETGT (position the file so the specified key is greater than the previous record key)
 - New control operation codes
 - a. OPEN (programmer control of when to open a file)
 - b. CLOSE (programmer control of when to close a file)
 - c. FEOD (force end of data to help control multiple logical files on a single device)
- *File operation codes* let you write a program without the requirement of a primary file. This provides a more straightforward approach to interactive applications and an alternative for complex batch programs.

- The following *new operation codes* are available for use with a work station device:
 - EXFMT (execute format, which does a WRITE/READ to a specific format)
 - WRITE (write a specific format)
 - READ (read a specific format)
- The following *operation codes* are available for use with the subfile capability:
 - CHAIN (read a record using relative record number)
 - WRITE (write a new record using relative record number)
 - UPDAT (update an existing record)
 - READC (read the next changed record from the subfile)
- Any *RPG III program* can call other programs. This allows parameters to be passed during the call and return of programs. Programs can be either RPG III, COBOL, or control language programs. New operation codes include:
 - CALL (call a program)
 - PLIST (parameter list)
 - PARM (parameter statement)
 - RETRN (return to the caller)
 - FREE (free the resources of a called program)

Also, a new indicator (RT) lets programs that are written by using the RPG III program cycle return to the caller.

- *Data structures* let you redefine a storage area one or more times, and let you process either the entire data structure or any of the subfields. This simplifies many application approaches and eliminates the coding and execution of many MOVE and MOVE* operations. Also, you can use an externally described data format to describe a data structure.

These data structures, called data areas (*DTAARA) in CPF, can exist in any library. RPG has access to any data area.

- *Multiple occurrence data structures* let you define a data structure and then repeat this definition any number of times in a program to form a series of data structures with identical formats. This eliminates the need for many arrays and provides a more logical approach to some programmer requirements.
- *Compare and branch operations (CABXX)* let you do a simple compare and branch in one operation, eliminating the need to set and test many indicators. The xx portion of the CABXX operation code may be GT, LT, EQ, NE, GE, LE, and UN (unconditional).

- *Short form of calculations* lets you use a simple ADD, SUB, MULT, or DIV operation where one of the factors can be implicitly specified. For example, the traditional statement:

FACTOR1	OPCODE	FACTOR2	RESULT
QTY	ADD	TOTQTY	TOTQTY

can be written:

FACTOR1	OPCODE	FACTOR2	RESULT
	ADD	QTY	TOTQTY

This eliminates unnecessary coding and makes programs easier to read.

- *EXCPT to a label* lets you control one or more exception output lines simply.
- *Simplified subroutine specification* eliminates the need to code SR in positions 7-8 of every subroutine statement. This eliminates coding and many diagnostics caused by coding errors.
- *Indicators as data* give you more indicator status control. This is helpful when communicating to other programs. In addition, the indicator array (*IN) containing indicators 01 through 99 eliminates coding and provides a more straightforward approach to many program processing requirements, such as the ability to set a block of indicators on or off.
- *DO loops* allow for automatic loop counting. The functions of do, do while (DOWxx), and do until (DOUxx) are supported. The do while and do until functions allow modifiers in the same manner as the compare and branch (CABxx) function. This can simplify programming where repetitive functions are required and also can provide an RPG III solution for users who want a structured programming approach.
- *IF/ELSE operations* let you execute a series of RPG III operations without the use of branching or indicator control. The IFxx operation lets you use modifiers as they are used in the compare and branch (CABxx) operation. These also provide an RPG III solution for the users who want a structured programming approach.
- *Program control of exception/error handling* lets the RPG III user control the exception/errors that can be raised by data management, system functions, and machine functions. This provides the user with options that let the program continue operation even though exception/errors have occurred.
- *TESTN operation code* lets you check the validity of an alphanumeric field to ensure that it contains only zoned decimal characters, blanks, or zoned-decimal characters with leading blanks.
- *DEFN operation code* lets you define fields by using attributes of another field. This provides more flexibility in the way programs are written so that fewer source code changes and considerations are required when programs are created and maintained.

- *Improved use of U1-U8 indicators* lets you set the U1 through U8 indicators on or off during a program. (The current status of the indicators can then be used by other programs.) The indicators can be automatically returned to the caller. This allows the conditioned execution of programs that follow or the passing of switches for use by following programs.
- *Relative end positions on output* allow the user to specify the output of a field based upon a relative position from the previously described field. This allows the user to define simple output records more easily when:
 - Each field is adjacent to the next.
 - A constant should follow a field with two intervening blanks.
 - Each field is separated by N blanks.
- *Resulting indicator on a move operation* lets you use resulting indicators on MOVE and MOVE* operations. This eliminates the need for additional operations to check for blank, zero, plus, or minus conditions.
- *Figurative constants* let you use RPG III-defined names to eliminate initialization statements, thus solving many program requirements. The RPG III names include:
 - *BLANK/*BLANKS—a field of all blanks
 - *ZERO/*ZEROS—a field of all zeros
 - *HIVAL—a field with the highest collating value
 - *LOVAL—a field with the lowest collating value
 - *ALL '...'—a field of an alphanumeric character string, which is used for cyclical functions
- *TIME operation code* lets the program retrieve the current time, or both the date and time, from the system timer. This lets you time stamp transactions and other functions.
- *SHTDN operation code* lets the program test an external event that requests the shutdown of the program.
- *SORTA operation code* lets you sort an array in either ascending or descending sequence.
- *KLIST/KFLD operation codes* let you specify the description of a key field that consists of multiple fields.
- *Special file with PLIST operation code* lets you pass parameters when the device is SPECIAL.
- *User-defined edit codes* let you define five edit codes, to handle unique editing.
- *Currency symbol* lets you specify symbols other than \$ for use in edit codes and edit words.
- *Dynamic space/skip function* lets your program dynamically specify space/skip operations instead of using literals on the output specifications.

- *Default control specification (header)* eliminates the need for having a control specification defined in every program and allows standardization of various functions such as editing and date format.
- *Table file replacement* lets you, at the end of the program, write a table/array back to the same file it was read from.
- *BITON/BITOF* of arrays let you use bit settings on whole arrays.

New Compilation Listing Functions

- *Cross reference list* shows you where fields and indicators are used and modified. This simplifies corrections and improves the documentation of the program.
- *Text description from data description specifications* lets you see any text descriptions provided for fields and indicators specified in the data description specifications. These descriptions appear on the listing as part of the program documentation.
- *Resulting indicator usage* lets you quickly determine which resulting indicator is being defined (the first, second, or third position). This eliminates confusion and results in better program documentation.
- *Consecutive position rule* lets you easily determine the exact position of any character in a source record.
- *Source entry utility sequence number* is used as the default numbering method for statements and diagnostics. This lets you go directly from a diagnostic that discusses a source statement in error to the specific statement using the source entry utility.
- *Source diagnostic error* notation lets you find errors easily by using position 1 of the compiler listing. Position 1 contains an asterisk for any error found. Asterisks are also placed under each field in error during the listing of the source.
- */EJECT and /SPACE specifications* let you skip lines in the source listing for easier reading of major program functions.
- *Nesting indication* lets you easily see which level of nesting a statement is included in for DO loops and IF/ELSE processing.

Auto Report

Auto Report is a standard feature of System/38 RPG III. It is executed as a separate program that produces source output and invokes the RPG III compiler. You can use the automatic formatting functions and /COPY to simplify programming requirements.

Enhanced Debug Facilities

- *Support of the system debug facility* lets you debug RPG III programs (interactively or by batch) without any special source statements. This facility includes such functions as:
 - Stopping at specific statement numbers or TAG labels, or at RPG III major functions (for example, the first detail calculation)
 - Displaying and changing fields, indicators, and arrays
 - Tracing specific ranges of statements and the output of changed fields
- *DEBUG operation code* does not require a file name for output. Output is automatically placed in the job log.
- *DUMP operation code* allows source statement control on when and how much information should be dumped.

Other Support

- An RPG source file (QRPGSRC) is provided by RPG III. Each RPG III source program is stored as a separate member of the source file. The user may create additional source files.
- A Create RPG Program (CRTRPGPGM) command generates RPG object programs. This command has various options for control of the source listing and security functions.
- A Create Auto Report Program (CRTRPTPGM) command invokes the RPG Auto Report program.

COBOL

System/38 COBOL is a powerful compiler that meets the December 1975 Federal Information Processing Standards (FIPS), low-intermediate, and the American National Standards (ANS) COBOL, X3.23-1974. It also has many features of the high-level FIPS except the communications and report writer modules.

IBM Extensions to COBOL

In addition to the standard language functions, System/38 COBOL provides the following extensions:

- *Use of apostrophe* in addition to quotes
- *Extended data types* of computational-3 (packed) and computational-4 (binary)
- *SET condition-name* to TRUE
- *SET mnemonic-name* to ON/OFF

Enhancements over System/3 COBOL

- *The COBOL SORT/MERGE verbs* are supported for multiple input files. Multiple sort and merge operations are allowed in the same program.
- *Nested IF statements* are supported.
- *ADD, SUBTRACT, and MOVE CORRESPONDING verbs* are supported. This enhancement reduces the number of instructions required to edit group items for printing or for the data conversion of group items.
- *Support for abbreviated combined relations* lets you code implied subjects and operands such that the following statement,

If A = B, or A = C...,

can now be abbreviated

If A = B, or C...

- *Substring capability* has been added so that you can use the verbs **STRING** and **UNSTRING** to combine or take apart fields, respectively. These powerful instructions are useful in text processing.

- *Table processing* has been enhanced to support the ASCENDING/DESCENDING options and the functions:
 - SEARCH (sequential search)
 - SEARCH ALL (binary search)
- *Variable-length tables* are supported by the OCCURS DEPENDING ON option, letting you restrict table searches to valid data only. This saves processing time.
- *The INSPECT verb* replaces the 1968-level EXAMINE verb. The function has been enhanced to allow the use of multiple characters.
- *File processing support* has been expanded in the following areas:
 - The DELETE verb option is supported for relative and indexed files.
 - Add to sequential files is supported through the OPEN with EXTEND option.
 - The START verb lets you do generic searches on indexed files.
 - Optional sequential files let you continue processing even if some of the files do not exist.
 - The ACCESS IS DYNAMIC clause is supported so that you can now process the same file both sequentially and randomly without having to define the file twice.
 - Through the use of the LINAGE clause option, you can specify top and bottom margins for a printer file.
 - The FILE STATUS function is supported so that you can interrogate the result of a file operation.
 - USE after STANDARD ERROR/EXCEPTION is supported to let you combine error/exception logic in one section of the program.
- *The value clause for level 88* supports multiple values. The THROUGH option lets you do range checking easily.
- *Multiple result fields* in arithmetic statements are supported.
- *The REMAINDER option* of DIVIDE is supported.
- *The RENAME capability* has been extended with:
 - Level 66 items
 - THROUGH option support
- *The edit character "/"* is now allowed for data edits.
- *The COPY function* has been enhanced to support the REPLACING option. This lets you replace copied character strings to fit the requirements of the source program.
- *The COPY function* lets you access externally described files by 'copying' DDS into the COBOL source program, creating COBOL data definitions.
- *The 1974 debug module* is supported. This gives you a special group item called DEBUG-ITEM that you can process at desired debug points in the program. This support also includes both compile and execution time switches to suppress the debug function.

Enhancements over System/34 COBOL

- *Diskette support.*
- *Magnetic tape support.*
- *96-column card support.*
- *A dynamic call and cancel function that permits improved flexibility in program-to-program communications.*
- *The USE FOR DEBUGGING feature includes support for filenames and identifiers.*

Other Support

- System/38 COBOL permits a COBOL program to be called by, or to call, a program written in System/38 control language, RPG III, or COBOL.
- A COBOL source file (QCBLSRC) is provided by System/38 COBOL for source programs.
- A Create COBOL Program (CRTCLPGM) command generates object programs. This command includes various options for control of the source listing and security functions.

Extensions Supporting Data Base and Work Station Processing

Using COBOL, you can define data as you did in the past. You can use Standard COBOL Environment and Data Division entries to specify file identification, field definitions, and data structures. System/38 has added phrases to the READ, WRITE, REWRITE, DELETE, and START verbs to support the System/38 data base and work stations.

System/38 COBOL supports numeric primary keys and noncontiguous composite keys for files assigned to DATABASE.

For COBOL programming, you also have the following new capabilities:

- You can program work stations, whether they are locally or remotely attached.
- You can use System/38 data base and display device definitions with your COBOL programs.

Programmer Services

System/38 provides the following functions to the COBOL programmer:

- Enhanced documentation, including *where used* information such as data base relationships, file/field definitions, and program references (for example, data name cross reference and verb usage list)
- Syntax checking of COBOL programs in either batch mode or interactively with the source entry utility (SEU)
- FIPS (Federal Information Processing Standard) Flagger, which makes it possible to ensure that the level of COBOL language used in a particular programming department conforms to a particular 1975 FIPS level

American National Standards (ANS) COBOL, X3.23-1974 is identical to ISO 1989-COBOL, approved in February 1978 by the International Organization for Standardization. System/38 provides the highest level for the ANS processing modules shown in Figure 2-12.

COBOL Processing Modules¹	Module Description
Nucleus 2 NUC 1,2	Contains the language elements that are necessary for internal processing.
Table Handling 2 TBL 1,2	Contains the language elements necessary for: (1) definition of tables, (2) identification, manipulation, and use of indexes, (3) reference to the items within tables. Provides the ability to define fixed-length or variable-length tables of up to three dimensions. Items in the tables can be referred to by using a subscript or an index.
Sequential I/O 2 SEQ 1,2	Allows definition and access of sequentially organized external files. ²
Relative I/O 2 REL 0, 2	Provides capability for defining and accessing disk files in which records are identified by relative record numbers. A file can be accessed randomly and sequentially in the same COBOL program. ²
Indexed I/O 2 INX 0, 2	Provides the capability for defining disk files in which records are identified by the value of a key and accessed through an access path (except for the ALTERNATE RECORD KEY clause and the KEY IS phrase of the READ statement). ²
Sort/Merge 2 SRT 0, 2	Allows the inclusion of one or more sorts in a COBOL program and use of the merge facility.
Segmentation 2 SEG 0,2	Provides the overlaying at object time of Procedure Division sections.
Library 2 LIB 0, 2	Allows inclusion of predefined COBOL text in a program.
Debug 2 DEB 0,2	Provides user specification statements and procedures for debugging.
Interprogram Communication 2 IPC 0, 2	Provides facilities for a program to communicate with one or more other programs. Also provides capability to transfer control to another program known at compile time, and the ability for both programs to have access to certain data items.
¹ The first digit represents the level of the modules included in the compiler; the second digit represents the lowest level specified for American National Standard COBOL (0 implies that the module may be completely missing from a standard compiler); the third digit represents the highest level specified in the standard. ² RERUN clause is treated as comments.	

Figure 2-12. COBOL Processing Modules and Descriptions

INTERACTIVE DATA BASE UTILITIES

The System/38 Interactive Data Base Utilities is a program product consisting of a comprehensive set of separate utilities. These utilities are:

- *Source entry utility* for creating and maintaining program language source files
- *Screen design aid* for creating and maintaining display device files and menus
- *Data file utility* for creating and maintaining data files and for displaying specific records from data files
- *Query utility* for extracting and presenting information from data files

Source Entry Utility (SEU)

The source entry utility helps you enter and maintain source statements for RPG III, COBOL, control language, data description specifications, and other standard IBM source formats.

Each statement or command can be syntax checked as it is entered. When control language source is entered, the command selection menus and the parameter prompting function can be used just as when performing command entry.

The major functions of the source entry utility help you:

- Add, delete, move, copy, and change statements in a source file member
- Copy statements from one source file member and include them in another
- Search a source file member for specific character strings
- Check syntax for statements or entire members
- Move statements from one place to another in a member
- Perform sequence numbering and control

Screen Design Aid (SDA)

The screen design aid (SDA) supplies an easy-to-use, interactive way for you to:

- Lay out a display format on the work station screen
- Test and verify a display format
- Rearrange and redesign a display format
- Generate the data description specifications (DDS)
- Create a display device file
- Create CL programs to support menus
- Remove a display format from a source file

Data File Utility (DFU)

The data file utility provides functions for data entry, file maintenance, and display of keyed data files.

DFU provides a series of menus and prompts that you can use to create, change, delete, maintain, and manage applications. You can use this menu interface to:

- Define and create a new application
- Change and recreate an existing application
- Define and create a new application using an existing application
- Execute an existing application
- Delete an existing application
- Change the descriptive text of an existing application
- Rename an existing application
- Move an existing application to another library

- Grant and revoke the authority of other users to owned applications
- Transfer the ownership of an application to another user
- Review names and information about files to which you are authorized

Some of the major functions of the data file utility let you:

- Create data file utility applications by (1) specifying the described data file to be used and (2) taking all the defaults. Therefore, you can create data entry, file maintenance, and inquiry applications, using the field definitions from the described data and a standard display format provided by the data file utility.
- Create an application to use only selected fields in the file.
- Define data file utility applications for files containing multiple record formats (logical files).
- Create data file utility definitions to be used for entering or displaying file data. The following alternatives are allowed:
 - Single display format used with single data base record types
 - Multiple display formats used with single data base record types
 - Single display format used with multiple data base records of the same type

The data file utility gives you:

- An optional audit trail of data entered in file maintenance operations
- A simple, direct method for retrieving specific data base records and for displaying the records in a specified format

Query Utility

The query utility lets you sequence, summarize, and tabulate data in a data base file. The results can be presented through a display or printed report. Query differs from inquiry primarily in that inquiry is essentially the selection of a single record, whereas the query utility selects a set of records, can influence the order of the set, and can generate summary reports.

Query provides a series of menus and prompts that you can use to create, change, delete, maintain, and manage applications. You can use this menu interface to:

- Define and create a new application
- Change and recreate an existing application
- Define and create a new application using an existing application
- Execute an existing application
- Display the output from a query at the work station
- Inquire into the status of submitted queries
- Delete an existing application
- Change the descriptive text of an existing application
- Rename an existing application
- Move an existing application to another library
- Grant and revoke the authority of other users to owned applications
- Transfer the ownership of an application to another user
- Review names and information about files to which you are authorized

During the query application definition phase, you can specify:

- Record selection
- Record sequencing
- Record sampling
- Result fields, which are computations based on fields
- Table definition, to allow ranking, averaging, summing, and tabulating
- Classification, tabulation, and presentation of data in two dimensions (matrix form)
- Report headers and titles
- Arrangement of fields on a report

TEXT MANAGEMENT PROGRAM RPO

The Text Management Program RPO lets you process text documents from any work station while the system is also being used for other applications. Text documents include forms, letters, procedures, and any other data that need to be stored for future printing. Text document processing includes originating, editing, personalizing, duplicating, line numbering, and page numbering documents, as well as putting running feet or heads on each page.

Text management lets you:

- Create text by keying in new text, by copying from stored documents, or by copying data from a data base file
- Store text as a document
- View data on a split screen as you edit, correct, add data to, or delete data from documents
- Revise text (by moving, copying, inserting, deleting, and shifting it) by using line commands and CF keys within the document's text
- Create a column list from a data base file
- Create, store, and retrieve forms and form letters, and complete the forms or individualize the form letters
- Control line spacing, the number of documents to be printed, and what portion of a stored document is to be printed
- Produce a program-generated table of contents for a document
- Print documents with formatting options for a head or a foot for each page, flagging changes, and printing line numbers for review copies

Text management lets you insert data from selected data base files into your documents. During creation or editing of a document, you simply key in the name of the data base field at the location where you want that data to appear. Then you specify the data base file from which this data will be extracted. Other data base access features of text management include inserting a list of data base data along with a document and performing browse and copy functions on data base data while your document remains on the top half of a split screen.

REMOTE JOB ENTRY FACILITY (RJEF)

The Remote Job Entry Facility (RJEF) lets System/38 function as a remote job entry work station to the host system while the System/38 submits jobs to a central System/370, 30XX, or 43XX. System/38 is attached to the central system by means of a point-to-point switched or nonswitched data link using binary synchronous communications (BSC). Central system RJE programming support should be generated specifying System/3 work station with console support.

System/38 RJEF communicates with the central processing subsystems OS/VS1 RES, OS/VS2 JES2, and OS/VS2 JES3 or VM 370 Remote Spooling Communications Subsystem (RSCS) as a System/3 RJE work station.

Commands to submit a remote job stream can be entered from a work station or as a batch job. Multiple submit requests can be queued and transmitted later on a first-in, first-out basis when the remote job entry facility is started.

Output from the central system can be directed to a printer device file (spooled or nonspooled), to a user prespecified data base file member, or to a data base file member optionally created by the RJEF communication process. The decision about where to direct the output is based on data in the forms mount message sent from the host system.

RJEF communications is supported for both attended and unattended execution.

CONVERSION REFORMAT UTILITY

The Conversion Reformat Utility operates upon data from a data base file, a diskette device file, or a card device file to perform sort, merge, and copy operations. The Conversion Reformat Utility lets you:

- Sort a physical file to produce a record address file
- Sort a physical file to produce a physical file
- Sort/merge multiple files to produce a physical file
- Copy data from one or more files to produce a physical file

Chapter 3. Machine Summary

System/38 is available in four models that offer a variety of input/output devices and a wide range of storage capacities.

This chapter describes the devices available. Figure 3-1 shows the various alternatives for System/38 machine configurations.

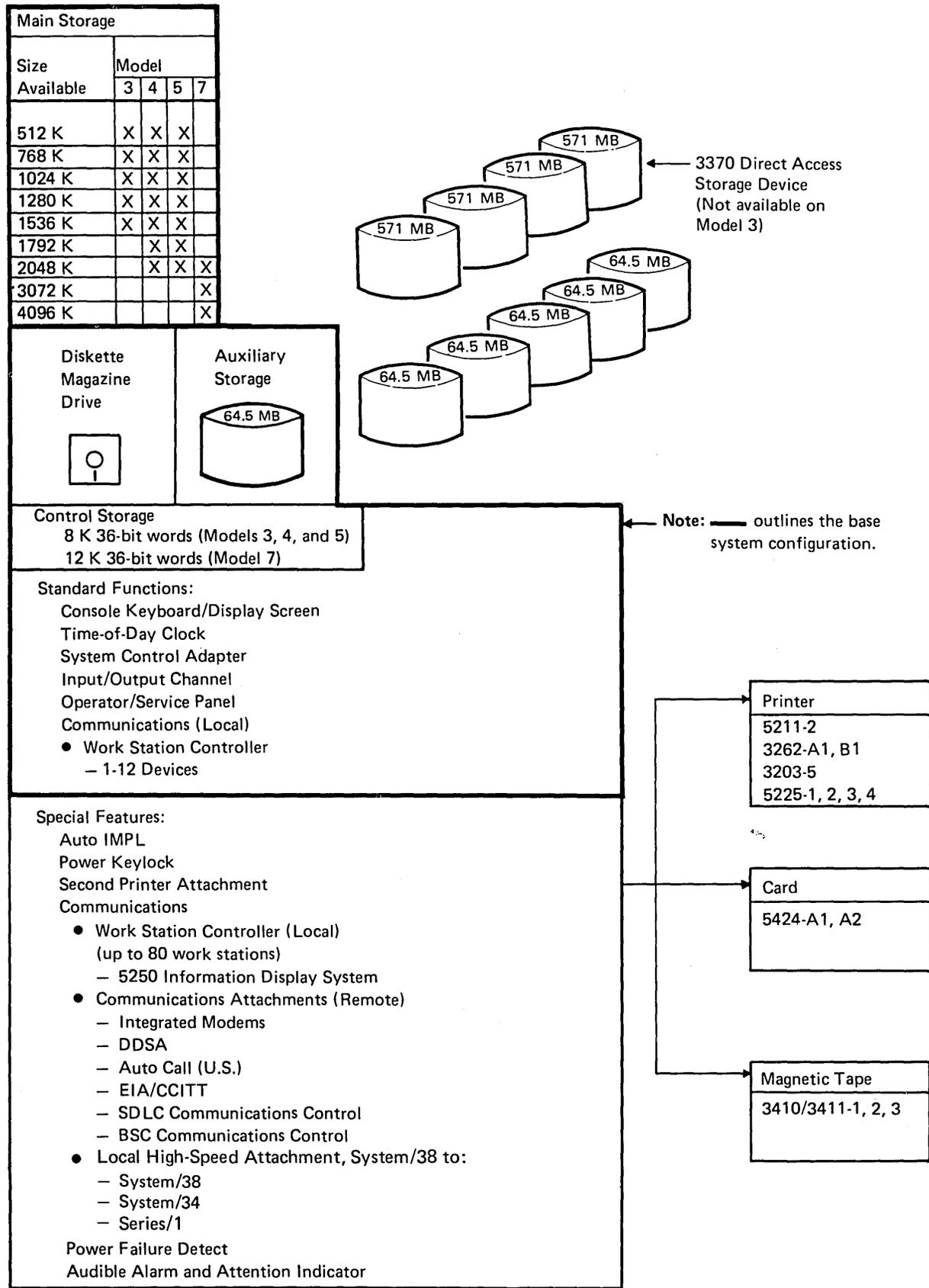


Figure 3-1. Machine Configurator

SYSTEM UNIT

By design, System/38 is oriented toward a multi-user system environment. Logical functions within the system unit are implemented in an advanced, high-density hardware technology. The major functional units of the IBM 5381 System Unit consist of the following:

- High-level machine interface
- The processing unit and its related control storage
- Storage management
- Virtual address translator
- Input/output channel
- Main storage
- Time-of-day clock
- Automatic initial microprogram load
- Power keylock

High-Level Machine Interface

The primary characteristic of System/38, identifying it as a major advance in computer system architecture, is its *high-level machine interface*. This interface embodies many of the basic supervisory and resource management functions usually found in control programs for traditional systems.

The object-oriented architecture is fundamental to the overall design of the powerful functions provided by System/38. Objects are structures such as programs, device descriptions, and data base files that are manipulated at a logical level through the system unit's instruction set. System/38 manages storage on an object basis, reducing your need to be concerned with main storage size, physical disk location, and internal implementation.

The high-level operations performed by the machine interface instructions give you the logical functions you need without dependence on their machine implementation. The power of these instructions is illustrated by data base operations that retrieve, update, and logically order data records.

The machine controls the access path to objects. This permits effective authority enforcement and automatic serialization of concurrent operations on the same object. Pointers, used to address objects, are tamper-proof to prevent the addressing of unauthorized objects. These features provide greater data integrity and security than are available on traditional systems.

Each machine user's program executes as an independent process. The machine also manages machine resources (processor, storage, devices) that are shared by processes. Interprocess communication is accomplished through queues and event signals. Locks can be applied to objects to control and serialize concurrent access to objects shared by several processes.

Programs are translated into an internal form before execution to achieve greater efficiency. Program variable attributes let the machine automatically perform data type conversions and allocate program work storage for these variables.

Input/output operations offer greater device independence through the use of the machine device support component and systems network architecture (SNA). The machine handles the intricacies of the channel, communication networks, and asynchronous device operations.

System/38 incorporates all these features, and more, into the machine. This high level of function is standard on every machine model regardless of storage size, processor type, or device configuration.

Processing Unit

The IBM 5381 System Unit (Figure 3-2) contains the processing unit, which is the control center of System/38, and supplies the facilities for:

- Fetching and storing data
- Arithmetic and logical processing of data
- Executing instructions in a particular sequence
- Controlling communication between main storage and the input/output devices

The 5381 is available in four models (3, 4, 5, and 7). The processing unit of Model 4 achieves about 1.35 times the internal performance level of Model 3; Model 5, about 1.75 times that of Model 3; and Model 7, about 3.50 times that of Model 3. Models 3, 4, and 5 each have 8 K 36-bit words of control storage; Model 7 has 12 K 36-bit words.

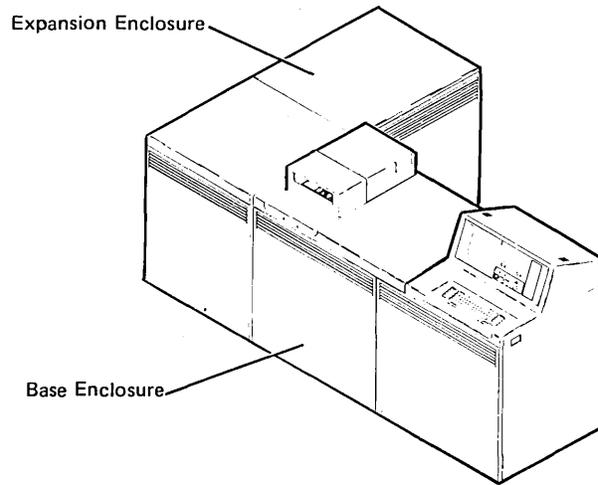


Figure 3-2. IBM 5381 System Unit with Expansion Enclosure

Storage Management

Program size depends very little on operating and design considerations, because main storage is not divided into partitions.

System/38 storage management offers such advantages as:

- Maintaining a directory of locations of all objects (such as files and programs) in the system. You no longer need to consider specific file location, disk volume, drive, or space requirements in the design of your programs.
- Allocating storage space as it is required. If more space is needed for a file during the execution of a program, the system obtains the space without interrupting the program. Also, as more storage capacity is added to the system, the system uses that space automatically, without requiring changes to existing programs or control language commands.
- Treating all auxiliary storage as a part of one continuous volume of space. This lets any file exist on multiple disk enclosures. A file is never limited in size by the amount of available space on a single disk enclosure.

Virtual Address Translator

The machine's virtual address translator is a machine function that converts virtual storage addresses to an internal form, thereby providing significant assistance to the storage management function.

Internal Input/Output Channel

The high-speed integrated channel has an instantaneous character transfer rate of up to 2.5 million bytes per second in byte mode, or up to 5.0 million bytes per second in halfword (that is, 16-bit, or 2-byte) mode. The 3370 Direct Access Storage Device attached to the 5381 operates in halfword mode. The other devices on or attached to the 5381 operate in byte mode. The channel connects the processing unit, main storage, and the input/output devices and permits the overlapping of input/output operations with processing.

Main Storage

Main storage is directly addressable storage for data and programs. Basic main storage consists of 512 K (524 288) bytes of storage and is equipped with error correction code hardware to automatically detect all single-bit and double-bit errors and to automatically correct all single-bit errors. Portions of storage with errors that cannot be corrected are bypassed, allowing system operations to continue. Main storage capacities available with the 5381 Processing Unit Models 3, 4, 5, and 7 are:

- 512 K (524 288) bytes (Models 3, 4, and 5 only)
- 768 K (786 432) bytes (Models 3, 4, and 5 only)
- 1024 K (1 048 576) bytes (Models 3, 4, and 5 only)
- 1280 K (1 310 720) bytes (Models 3, 4, and 5 only)
- 1536 K (1 572 864) bytes (Models 3, 4, and 5 only)
- 1792 K (1 835 008) bytes (Models 4 and 5 only)
- 2048 K (2 097 152) bytes (Models 4, 5, and 7 only)
- 3072 K (3 146 728) bytes (Model 7 only)
- 4096 K (4 195 304) bytes (Model 7 only)

The data path between main storage and the processing unit is 4 bytes wide. The average main storage cycle time is approximately 1100 nanoseconds for Models 3 and 4, 600 nanoseconds for Model 5, and 400 nanoseconds for Model 7.

Time-of-Day Clock

The time-of-day clock is a timer that runs continuously during normal modes of operation. As a user, you can initialize and read the time-of-day clock.

Automatic Initial Microprogram Load (Auto-IMPL)

If commercial power is interrupted while Auto-IMPL is on, this special feature automatically initiates a power-on sequence after commercial power is restored. Once the Control Program Facility has been loaded into the system, this feature lets a recovery program that you have written resume operations. (The system operator controls this feature through the Auto-IMPL switch.)

Power Keylock

This special feature supplies a key-controlled switch, in series with the Power switch, to restrict the power-on cycle to employees who have a key.

Power Warning Feature

When an uninterruptible power supply (UPS) is attached to the System/38 power source, the power warning special feature provides a signal to System/38 whenever the input AC voltage falls below an acceptable operating level. The Control Program Facility uses this signal to bring the system to an orderly halt. (For more information about the UPS, see *Uninterruptible Power Supply Selection and Installation Planning Guide*, GA21-9421.)

Audible Alarm and Attention Indicator

This special feature provides an attention indicator and an audible alarm with a volume control on the console. The indicator and alarm alert the system operator about a waiting message that requires attention.

SYSTEM CONSOLE

The System/38 console (Figure 3-3) includes a display screen, a keyboard, and an operator/service panel. The console is used primarily for interactive communication between the system operator and the programs (Control Program Facility and user programs) executing in the system, and between the system operator and the work station users.



Figure 3-3. System Console

System Console Display Screen

The system console display screen (Figure 3-4) can display up to 16 lines of information, having 64 characters per line. The characters on the console display screen are larger than those displayed on the work station display screen. This improves readability for the operator who may be some distance away. The operator can change the display intensity by using a control on the operator/service panel.

In addition to the standard attention indicator, an optional feature provides a backlighted indicator, an audible alarm, and a volume control. These features alert the operator to pending messages that require attention.

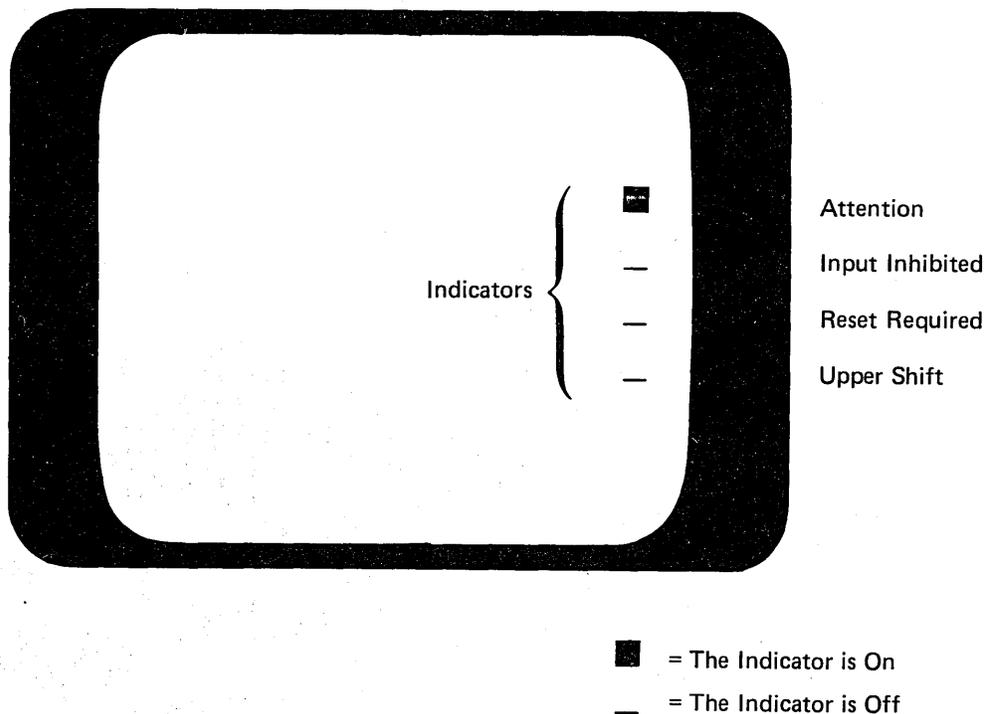


Figure 3-4. System Console Display Screen

The system console can be used for:

- System operator functions
- Programmer functions
- Application functions (similar to a work station)
- Maintenance functions (by the service representative to run tests and to display diagnostic information)

System Console Keyboard

The system operator uses the system console keyboard (Figure 3-5) when entering control language commands and when responding to menus, prompts, and messages for operator action.

The keyboard contains:

- Standard alphameric keys (letters, numbers, and special characters) for entering alphabetic (uppercase and lowercase) and numeric data
- Cursor movement keys to position the cursor on the display screen
- Easily identified command function keys (CF1 through CF24) for requesting specific program functions



Figure 3-5. System Console Keyboard

System Console Operator/Service Panel

The operator/service panel (Figure 3-6) has lights, switches, and keys that are used for communication between the system operator (or service representative) and the system.

- The lights indicate conditions in the processing unit or the system.
- The major switches are used to power on the system and initiate the microprogram load process. The service representative can use additional switches to control system operations.

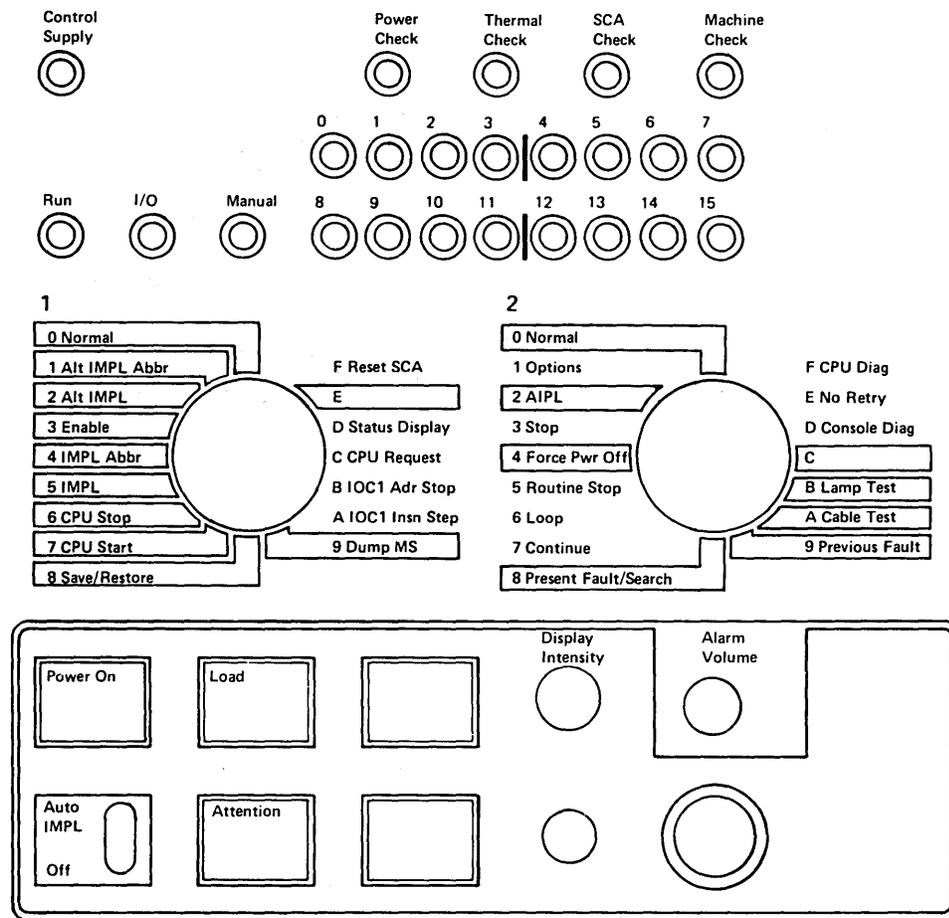


Figure 3-6. System Console Operator/Service Panel

System Control Adapter

The system control adapter (although not visible to the user) is the interface between the system and the operator/service panel. Whenever the system is initially powered on, the system control adapter provides an automatic diagnostic check of the major units of the system unit (such as auxiliary storage, diskette magazine drive, the system console, and the processing unit).

The system control adapter permits:

- Servicing of input/output devices (not included in the base system; see Figure 3-1) while processing continues, if the System/38 Control Program Facility is installed
- Diagnostic analysis of the processing unit, main and auxiliary storage, and the device adapters
- Altering and displaying processing unit registers
- Logging error conditions and stopping the system
- Stopping the system and supplying a synchronous pulse when the system encounters a specified address

AUXILIARY STORAGE

System/38 can be equipped with 64.5 through 2672.2 megabytes of magnetic disk media auxiliary storage. Auxiliary storage consists of (1) the data storage function of the system unit and (2) IBM 3370 Direct Access Storage Devices.

System Unit Data Storage Function

Each System/38 is equipped with at least one, and can be equipped with one through six, nonremovable disk enclosures, which provide a data storage function. Disk enclosures 1 and 2 are packaged in the system unit base enclosure; disk enclosures 3, 4, 5, and 6 are packaged in the system unit expansion enclosure (Figure 3-7).

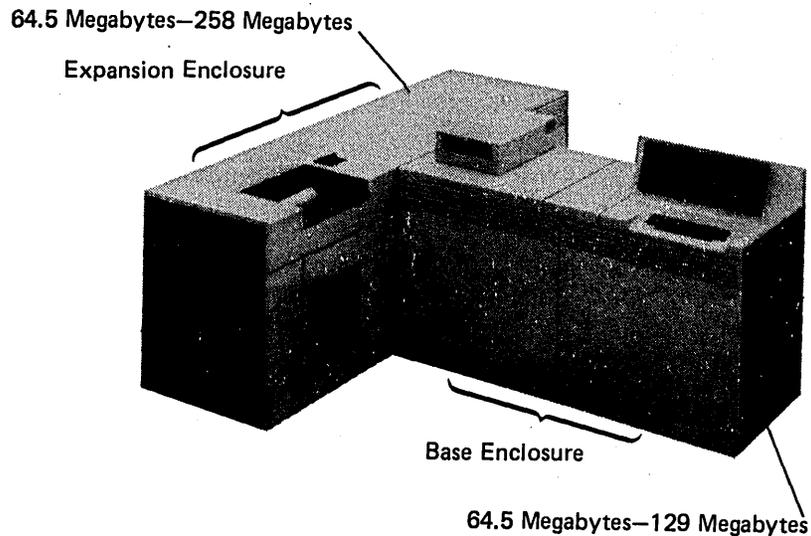


Figure 3-7. Data Storage Function Disk Enclosure Locations

The performance and physical characteristics of the data storage function are:

- Seek time: 25 milliseconds (average)
- Latency: 9.6 milliseconds (average)
- Data transfer rate: Up to 1 031 000 bytes per second
- Disk specifications:
 - Number of sectors: 33 per track (0-32)
(One sector, normally unassigned, is for alternative use; that is, if a sector in a track fails, its use is assigned to the same track in the alternative sector.)
 - Size of sector: 512 bytes of data
- Capacity: 64 520 192 bytes

3370 Direct Access Storage Device

The IBM 3370 Direct Access Storage Device (Figure 3-8), offered as an auxiliary storage special feature on System/38 Models 4, 5, and 7, consists of one Model A11 storage and control unit (both units are in the same frame) and up to three additional Model B11 storage units, for a maximum of four units.

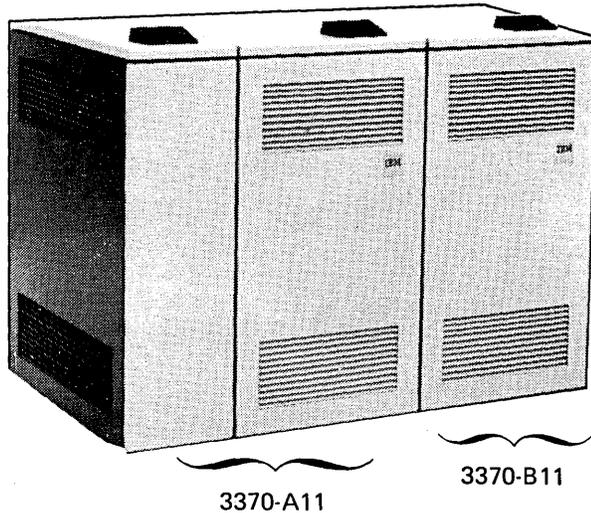


Figure 3-8. IBM 3370 Direct Access Storage Device

Each Model A11 and B11 supplies 571 megabytes of storage on nonremovable magnetic disks. The four units provide a total of 2285 megabytes of additional auxiliary storage. The performance and physical characteristics of the 3370 are:

- Seek time: 20 milliseconds (average)
- Latency: 10.1 milliseconds (average)
- Data transfer rate: Up to 1 859 000 bytes per second
- Disk specifications:
 - Bytes per block 512
 - Blocks per actuator position 744
 - Blocks per actuator 558 000
 - Bytes per actuator 285 696 000
- Capacity: 571 392 000 bytes

DISKETTE MAGAZINE DRIVE

The diskette magazine drive (Figure 3-9) is standard on all models of System/38. It is used primarily for the following functions:

- Save/restore—The selective backing up of data, programs, or files kept online in auxiliary storage. It also permits easy reloading of data files, if required, for recovery.
- Diskette input/output—The interchange of data where diskettes are exchanged with other systems and devices, such as System/32, System/34, IBM 5110 computer, and IBM 3741 data stations.
- System servicing—The loading of system diagnostic programs, engineering changes supplied to the service library, program changes, and new program releases from the IBM Program Information Department (PID).

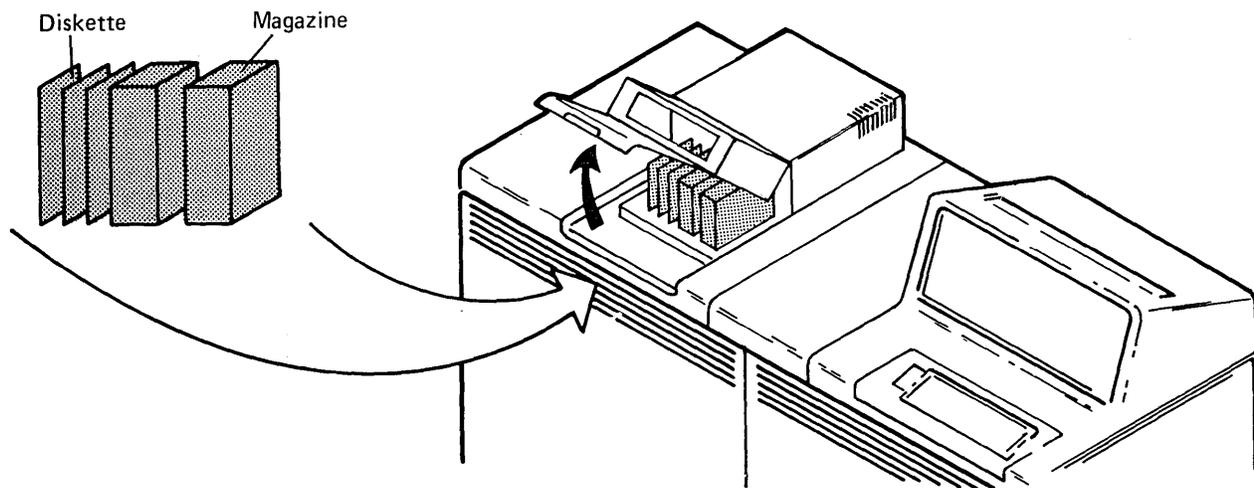


Figure 3-9. Diskette Magazine Drive

The diskette magazine drive is a sequential save/restore device that reads and writes data, at a high speed, from and to a diskette. A diskette is a small, flexible, magnetic disk that is permanently enclosed in a protective jacket. Each diskette can be removed from the diskette drive.

The following are the diskette types and their formats:

Diskette Type	Bytes per Sector	Diskette Capacity
Diskette 1 (one-sided)	128	246 272 bytes
	256*	284 160 bytes
	512*	303 104 bytes
Diskette 2 (two-sided)	128	492 544 bytes
	256*	568 320 bytes
	512*	606 208 bytes
Diskette 2D (two-sided, double-density)	256	985 088 bytes
	512*	1 136 640 bytes
	1024	1 212 416 bytes

*Supported at the machine-interface level only

For more information about the IBM diskette, see the *IBM Diskette General Information Manual, GA21-9182*.

The diskette magazine drive has two magazine positions and three individual diskette slots. Each magazine can contain up to 10 diskettes, resulting in a total capacity of 23 diskettes.

The individual diskette slots are generally used for small quantities of data (for example, when entering data from individual diskettes received from 3741 data station operators). For large quantities of data, the 3741 data station operators can load their diskettes into a magazine. The System/38 system operator can then load up to 10 diskettes in one physical operation. Typically, processing begins with the first diskette in magazine and automatically proceeds, diskette by diskette, through both magazines. Processing can be interrupted to replace a magazine. Then processing can continue again with the new magazine available for reading.

The maximum capacity of the diskette magazine drive containing two full magazines (using diskette type 2D and formatted in 1024-byte records) is approximately 24 megabytes. The performance of the diskette magazine drive varies depending on the amount of work the system has been given to do and the quantity and length of the data files involved.

Advantages of the diskette magazine drive include:

- Less operator intervention—Up to 24 megabytes of data may be written on two magazines before system operator action is required.
- Improved operator convenience—With the diskette magazine, diskette handling, labeling, and storage are simplified.
- Multifunction device operations.
 - Save/restore
 - Spooling
 - Copy
 - RPG III file operations
 - COBOL file operations

PRINTERS

You can assign a maximum of two system printers to System/38. System/38 supports the following printers as system printers:

Type	Model	Rate (Lines per Minute)
5211	2	300*
3262	A1,B1	650*
3203	5	1200*
5225	(see <i>Work Station Printers</i>)	

*Rates are based on a 48-character-set print belt or train cartridge.

The 3262 Model A1 must be bolted onto the system unit, thereby becoming an integrated part of System/38. (As a bolt-on printer, Model A1 shares power with the system unit.) All other printers are stand-alone printers with their own power supplies. Any combination of two system printers is allowed, except that you cannot install a 3262-A1 printer as the second printer.

All the system printers have improved setup features and quiet operation. The quiet operation is achieved by feeding paper completely within the covers. Forms control, as well as the selection of 6 or 8 lines per inch spacing, can be programmed.

Each of these printers is a line printer; that is, each printer can print one complete line at a time, up to 132 characters per line. Rates vary, depending on character set size, the number of characters, and the form skips associated with the output to be printed. The printers can print an original copy with up to five carbon copies, on paper from 3.5 inches to 15.25 inches wide (16 inches wide on the 3262 and 20 inches wide on the 3203), and from 3 inches to 14 inches long (24 inches long on the 3203).

The 3262 Printer looks like the 5211 Printer (Figure 3-10).

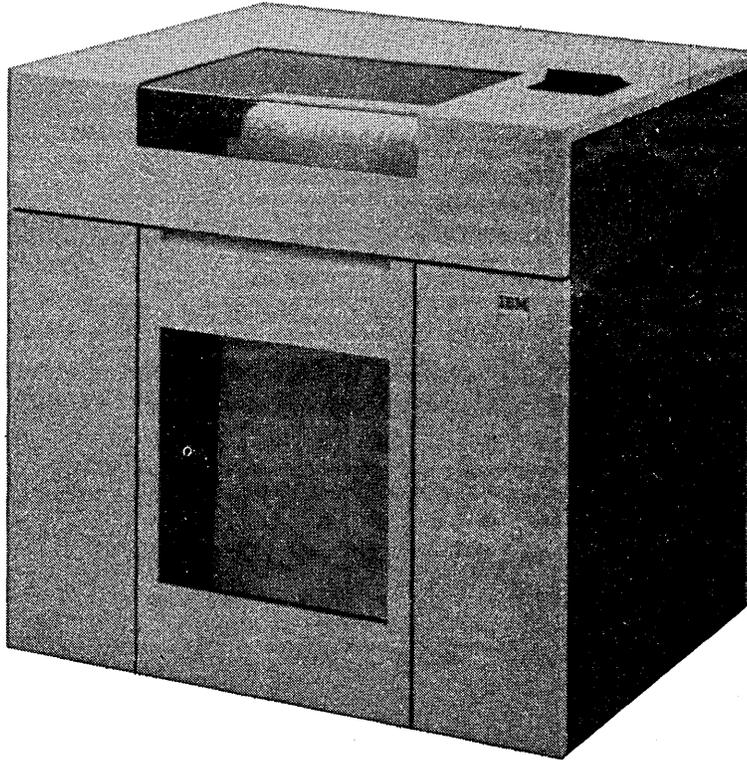


Figure 3-10. 5211 Printer

The 5211 and 3262 printers use a flexible steel print belt that the operator can change easily. These inexpensive belts allow the System/38 user to get a range of character sets and print speeds.

Specialized print belts that can yield higher print rates are available for the 5211 and 3262 printers.

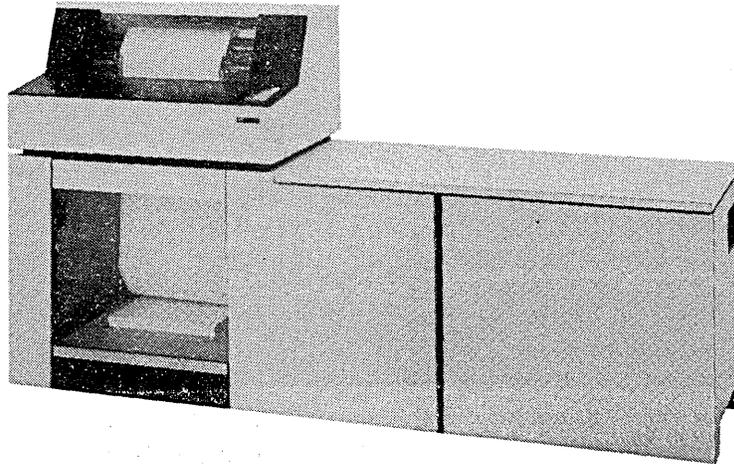


Figure 3-11. 3203 Printer, Model 5

The 3203 Printer (Figure 3-11) is a stand-alone unit attached to System/38 through interface cables.

The 3203 Printer uses an IBM 1416 Interchangeable Train Cartridge that the operator can change easily.

5211 Printer

A character set buffer of 192 positions in the 5211 printer attachment allows use of graphic sets of up through 192 characters. Print rates vary depending on character set size and frequency of character repetition on the belt. General-purpose belts of 48, 64, and 96 characters are available; they offer the following nominal rates:

Number of Characters	Rate (Lines per Minute)
48	300
64	235
96	164

- The 38-character set belt contains 38 graphics repeated around the belt. With this belt, the following rate can be attained:

Model	Rate (Lines per Minute)
5211-2	355

- The optimized numeric belt contains 42 graphics and is optimized to the printing of numbers, the period, and the comma. When printing single-spaced numeric information, the following rate can be reached:

Model	Rate (Lines per Minute)
5211-2	395

When printing the graphics A through Z with the 42-character set belt, the following rate can be reached:

Model	Rate (Lines per Minute)
5211-2	300

3262 Printer

A character set buffer of 288 positions in the 3262 printer attachment allows use of graphic sets of up through 288 characters. Standard print belts of 48, 64, and 96 characters are available; they offer the following nominal rates:

Number of Characters	Rate (Lines per Minute)
48	650
64	467
96	364

A general-purpose optimized 64-character set print belt is available for the 3262 Printer. This print belt can improve the performance of the printer over the standard 64-character print belt. Print chains containing OCR A-font and OCR B-font characters are available for magnetic ink character recognition applications.

60-Character System/38 Special Belt

The 60-character special belt lets you print the System/38 control language characters.

The 60-character System/38 special belt is available on the 5211 Printer and the 3262 Printer.

This belt is specially designed such that the user of a 5211 or 3262 Printer needs only one belt to print the usual commercial print and characters and the characters allowed by the control language syntax.

The System/38 control language requires more than 48 characters; consequently, you will need multiple belts unless you use the 60-character System/38 special belt.

3203 Printer

A character set buffer of 240 positions in the 3203 printer attachment lets you use graphic sets containing as many as 240 characters. One IBM 1416 Interchangeable Train Cartridge is required for each 3203 Printer. Standard train cartridges of 48, 60, and 120 characters are available; they offer the following nominal rated speeds:

Number of Characters	Speed (Lines per Minute)
48	1200
60	1017
120	581

Several special train cartridges are available for the 3203 Printer.

The 3203 Printer carriage has two carriage speeds: 24 inches per second for skipping from one to six lines, and 55 inches per second for skipping more than six lines.

The 3203 Printer uses powered feed rollers in the automatic forms stacking device to ensure rapid stacking of forms.

OCR (optical character recognition) print quality is standard on the 3203 Printer, although a special ribbon must be used for OCR printing. Therefore, documents printed by the 3203 can be used as input documents for OCR readers, eliminating manual handling that might otherwise be needed. Although designed primarily for OCR applications, the special ribbon can be used for other applications where enhanced print quality is important.

CARD DEVICE

5424 Multi-Function Card Unit

The IBM 5424 Multi-Function Card Unit (Figure 3-12) is offered on System/38.

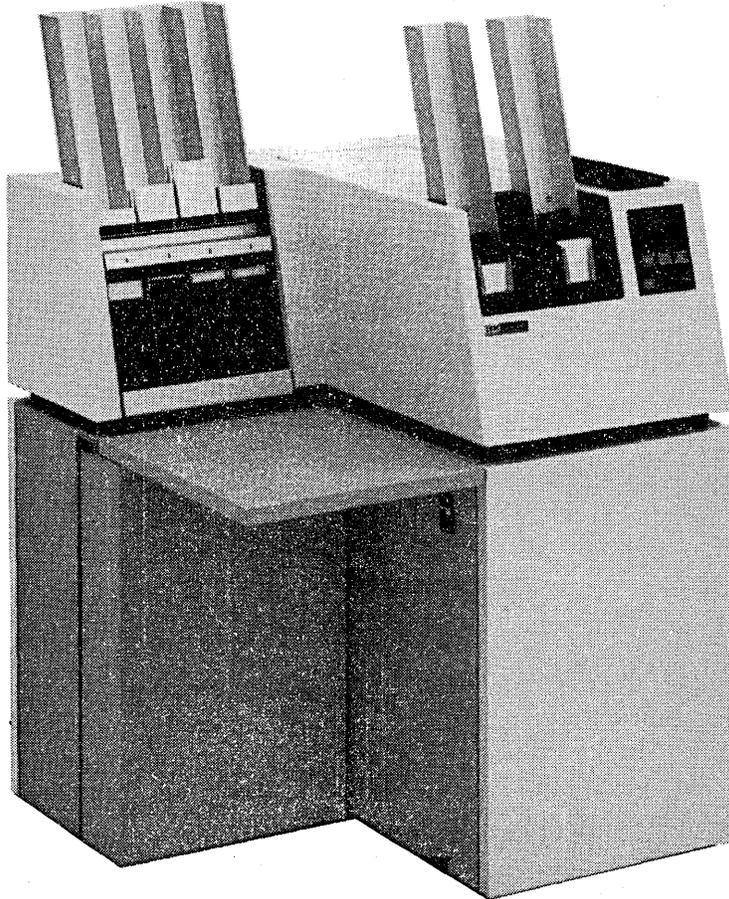


Figure 3-12. IBM 5424 Multi-Function Card Unit

As an input device, the 5424 can read 96-column cards from either of two hoppers and can stack cards in any of the four stackers.

The 5424 can read information from a card; then print, punch, or print and punch information onto the same card during a single card cycle.

As an output device, the 5424 can punch, print, and stack cards in any of the four stackers.

Two models of the 5424 are available:

Model A1

Reads 250 cards per minute

Punches 60 cards per minute

Prints 60 cards per minute¹

Model A2

Reads 500 cards per minute

Punches 120 cards per minute

Prints 120 cards per minute¹

MAGNETIC TAPE DEVICE

3410/3411 Magnetic Tape Subsystem

The IBM 3411 Magnetic Tape Unit and Control is offered as a tape drive on System/38. One of three IBM 3410 Magnetic Tape Units (Figure 3-13) can also be attached to the system; these are controlled by the 3411.

There are three models of the 3410/3411 subsystem:

- Model 1 moves tape at 12.5 inches per second, resulting in a maximum data rate of 20 000 bytes per second
- Model 2 moves tape at 25 inches per second, resulting in a maximum data rate of 40 000 bytes per second
- Model 3 moves tape at 50 inches per second, resulting in a maximum data rate of 80 000 bytes per second

Note: Interconnected 3410/3411 units must be the same model. Models cannot be intermixed.

The 3410/3411 Magnetic Tape Units can operate in the single-density or dual-density mode:

- 1600 bits per inch, phase-encoded, single-density
- 800 bits per inch, NRZI (non-return-to-zero change-on-ones recording), dual-density
- 1600 bits per inch, phase-encoded, dual-density

Both the 3410 and the 3411 are table-high units with tape reels installed horizontally rather than vertically. A transparent sliding cover permits easy access to the tape reels.

¹Assuming three lines of print.

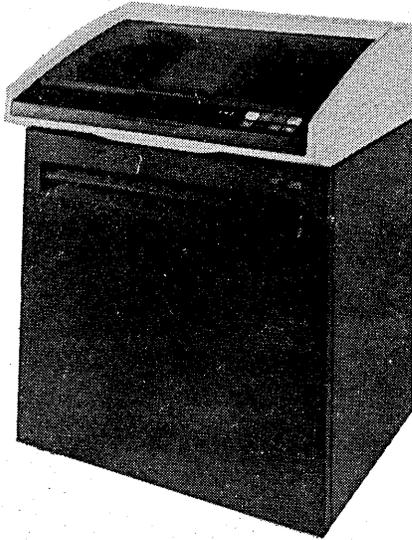


Figure 3-13. IBM 3410 Magnetic Tape Unit

COMMUNICATIONS

System/38 can communicate with multiple work stations at local and remote locations. Work stations can operate concurrently with no apparent interference (dependent upon the system work load) in the operation of any other application or individual work station.

Two attachment methods exist (Figure 3-14): one is through the work station controllers, the other through the communications attachment.

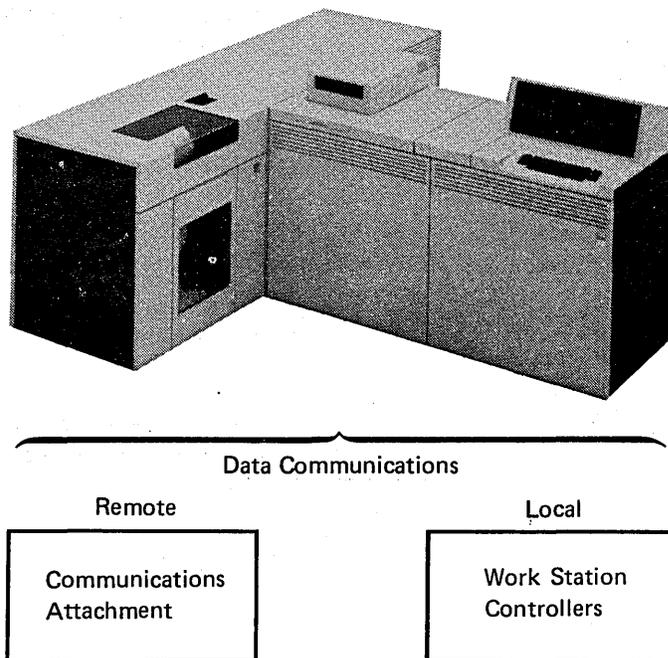


Figure 3-14. Local and Remote Communications Attachment

Work Station Controllers

The work station controllers provide for the local attachment of IBM 5250 Information Display System devices.

Up to 80 work stations can be connected directly to System/38 through this attachment method. The following work stations can be attached:

- 5251 Display Station Model 11
- 5224 Printer Models 1 and 2
- 5225 Printer Models 1, 2, 3, and 4
- 5256 Printer Models 1, 2, and 3

There is no need for common-carrier-provided services to attach the work stations to the work station controller. The limitations of telecommunication data rates do not pertain to the work station controller.

Basic Work Station Controller

The first work station controller is *standard* on all models of System/38 and provides eight ports for attaching work stations. These eight ports permit the direct attachment of up through 12 work stations (Figure 3-15). The maximum number of work stations per port is seven.

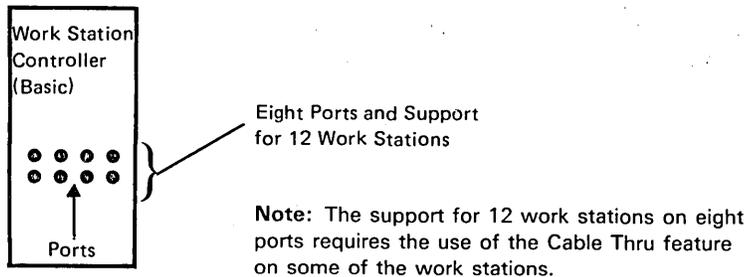
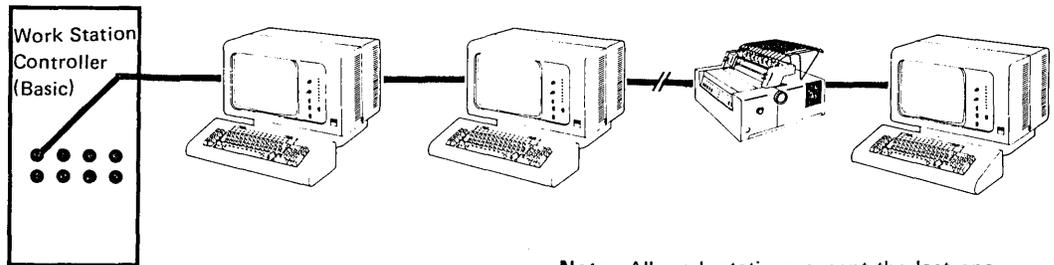


Figure 3-15. Work Station Controller (Basic)

Twinaxial Cable

Two-wire (twinaxial), shielded cable is used for attaching the work stations. It allows a maximum total cable length of 5000 feet (1525 meters) and multiple work stations (seven) to be attached to a single port. Figure 3-16 shows an example of attaching work stations on one twinaxial cable. This cabling arrangement requires use of the 5250 Cable Thru feature.



Note: All work stations except the last one are equipped with the Cable Thru feature.

Figure 3-16. Example Using Twinaxial Cable

Device Control Expansion

The Device Control Expansion special feature lets the work station controller support up to eight additional work stations. This feature does not provide any additional ports, but provides the necessary control storage to support eight additional work stations attached through the ports supplied with the work station controller (Figure 3-17).

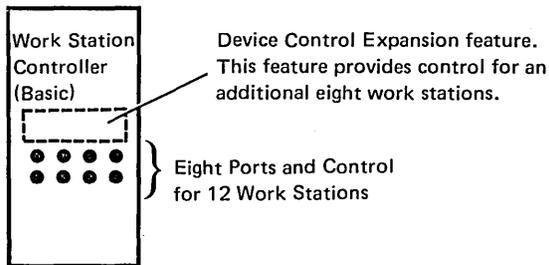


Figure 3-17. Device Control Expansion

Note: This feature cannot be installed with the Device Interface Expansion feature.

Device Interface Expansion

The Device Interface Expansion special feature provides the necessary control and eight more ports for the attachment of additional eight work stations (Figure 3-18).

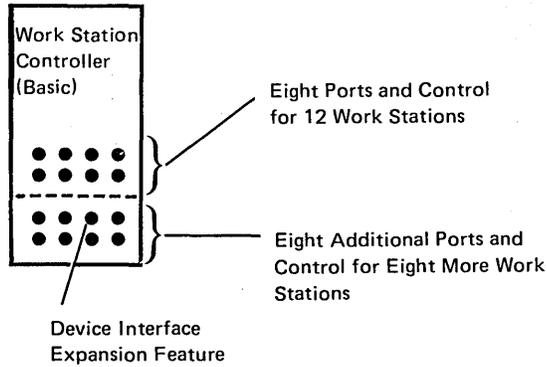


Figure 3-18. Device Interface Expansion

Note: This feature cannot be installed with the Device Control Expansion feature.

Second, Third, and Fourth Work Station Controllers

Each additional work station controller is a special feature available on all models of System/38. Each controller provides support that is identical to that of the basic work station controller. For attachment of additional work stations, see the discussions about the Device Control Expansion and the Device Interface Expansion special features earlier in this chapter.

Communications Attachment

Two communications attachments provide the necessary system control for the direct attachment of up to eight remote communications lines. The communications attachments let you attach devices that require data transmission services (such as private or common-carrier-supplied lines).

Voice grade transmission services at 600, 1200, 2000, 2400, 4800, 7200, and 9600 bits per second (bps) across private or common-carrier lines are supported.

Current analog-type data communications facilities and also the Digital Data Service Adapter for the American Telephone and Telegraph Company Data-Phone¹ Digital Service (U.S. only) are supported. Analog-type facilities are supported through the integrated modems (under the system's covers) at 600, 1200, 2000, 2400, or 4800 bps, or through external modems attached through the EIA/CCITT interface features. Digital facilities use no modems and are supported at 2400, 4800, or 9600 bps through the Digital Data Service Adapter.

Synchronous Data Link Control (SDLC) Support

Through this communications support, System/38 can communicate as a terminal system to IBM System/370, CICS/VS, and IMS/VS using synchronous data link control (SDLC) and systems network architecture (SNA).

System/38 communicates with IMS/VS applications executing under OS/VS1 or OS/VS2 (MVS) as well as CICS/VS applications executing under DOS/VSE, OS/VS1, or OS/VS2 (MVS) in System/370 (Models 145 through 168 for IMS/VS and Models 135 through 168 for CICS/VS and 3031, 3032, 3033, 4331, and 4341 processors). Communication with any of these operating systems while running under VM/370 is also supported. Communication is with levels of VTAM, ACF/VTAM, ACF/VTAME, TCAM, or ACF/TCAM supported by IMS/VS or CICS/VS at the time of shipment. NCP/VS or ACF/NCP/VS will be required as appropriate.

¹Trademark of American Telephone & Telegraph Co.

Binary Synchronous Communications (BSC) Support

System/38 binary synchronous communications, using RPG III or COBOL programs, can communicate with the following devices over point-to-point or multipoint communications facilities:

System or Device Attached to System/38	Communications Support Used
IBM Series/1	- RPS and EDX
IBM System/32	- RPG II TCF
IBM System/34	- RPG II TCF
	- SSP-ICF (BSCEL - as a System/3)
IBM System/3	- RPG II TCF
	- MLMP
	- CCP
IBM System/38	- RPG III and COBOL
IBM System/370, 30XX, 43XX	
DOS/VSE	- BTAM with EP/VS
	- CICS/VS
OS/VS1	- BTAM with EP/VS
	- TCAM or ACF/TCAM with NCP/VS, ACF/NCP/VS, or EP/VS
	- CICS/VS
OS/VS2	- BTAM with EP/VS
	- TCAM or ACF/TCAM with NCP/VS, ACF/NCP/VS, or EP/VS
	- CICS/VS

Note: System/370 Models 115 through 168, 30XX, and 43XX processors require one of the following: ICA on Models 115 through 138, 4331, or 370X with NCP/VS, ACF/NCP/VS, or EP/VS.

IBM 5280	- DE/RPG, COBOL
IBM 5110/5120	- APL, BASIC

Note: System/38 supports each of the following as a terminal (3741 BSC protocol):

IBM 3741 Data Station/Programmable Work Station
IBM 5110/5120 Computing System
IBM 5230 Data Collection System
IBM 5260 Retail System
IBM 5280 Distributed Data System
IBM 5520 Administrative System
IBM Office System/6
IBM 6240 Communicating Magnetic Card Typewriter
IBM 6580 Displaywriter System
IBM 6640 Document Printer
IBM 6670 Information Distributor
IBM CMCII Communicating Magnetic Card Typewriter II

Local High-Speed Lines for System Data Exchange

To provide high-speed data transfer between systems (at nominal rates of up to 56 000 bits per second), System/38 can be connected with another system as follows:

- System/38 to another System/38 (80 feet maximum between systems)
- System/38 to System/34 (80 feet maximum between systems)
- System/38 to Series/1 (400 feet maximum between systems)

Data transfer is point-to-point and program-to-program using the binary synchronous communications program support.

A maximum of one high-speed line is allowed for each communications attachment installed on the System/38, for a total of two lines per System/38. While a high-speed line is being used, no other line connected to the same attachment can be used.

5250 Information Display System

The 5250 Information Display System contains printers and tabletop display stations that are used as work stations. These work stations can be attached to System/38 by means of either the System/38 work station controller or the System/38 communications attachment feature. The 5250 system consists of the following devices, in various configurations:

- 5224 Printer¹
- 5225 Printer¹
- 5256 Printer¹
- 5251 Display Station Model 11¹
- 5251 Display Station Model 12²

The 5250 devices display and print both uppercase and lowercase alphabetic characters. Blink, column separator, high intensity, nondisplay, reverse image, and underscore functions are available for the 5250 display stations.

¹This device can be attached (locally) to the work station controller feature in the System/38 system unit or (remotely by means of a cluster feature) to a 5251 Model 12.

²This work station contains its own controller and must be connected to System/38 by means of a communications network attached to the System/38 communications attachment feature in the system unit.

5251 Display Station

The following characteristics apply to models of the 5251 (Figure 3-19).

- A display screen with 1920 characters (24 lines of 80 characters).
- A movable keyboard with three key arrangements on the 5251.
- A keyboard command key for selecting up to 24 additional, programmable, command functions.
- (5251 Model 12 only) An internal communications adapter for the remote attachment to a common carrier communications line using the SDLC line protocol with:
 - Transmission speeds of up to 9600 bits per second.
 - Two cluster features that allow the direct attachment of up to eight printers and display stations (except 5251 Model 12), in any combination.

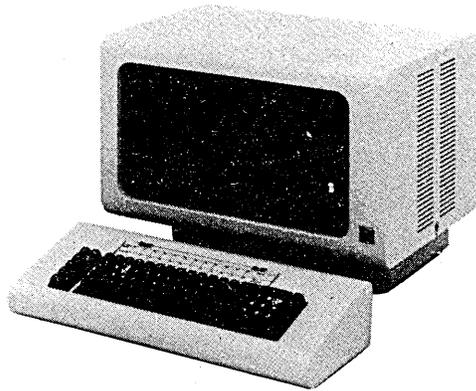


Figure 3-19. 5251 Display Station

Work Station Printers

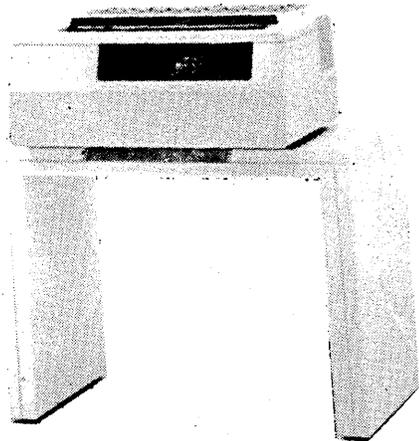
The printers described in Figure 3-20 can be attached either to (1) the System/38 work station controller as local work stations or (2) an IBM 5251 Model 12 as remote work stations. (The 5251 Model 12 must also be connected to a communications attachment on System/38 by communications facilities such as a telephone network or a customer-owned line.)

These printers (Figure 3-21) can be used for two different functions:

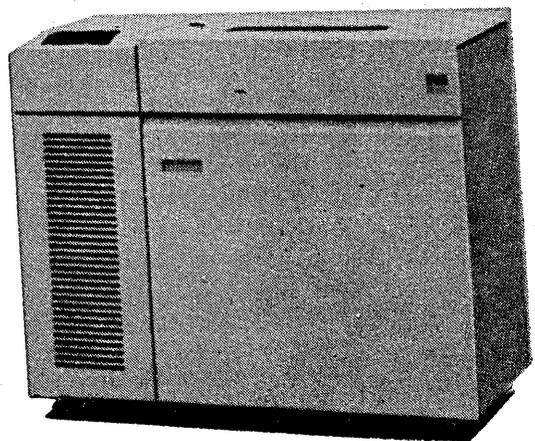
- They can print information and messages received from the system.
- They can print information displayed on the screen. Information displayed on the screen may have been received from the system or may have been entered by the work station operator. Both the work station operator and the system can control what is printed on the work station printer.

Name	Model	Max Char/Line		Vertical Line Spacing (Lines/In.)	Type of Forms Used	Type of Printing	Print Rate
		10 Char/In.	15 Char/In.				
5224 Printer	1	132	198	4, 6, 8, or 9	3.0- to 17.7-in., 1-through 4-part continuous forms	Wire matrix line	140 lines/min. @ 10 char/in., 95 lines/min. @ 15 char/in. (assuming 7.4 in. lines)
	2	132	198	4, 6, 8, or 9	3.0- to 17.7-in., 1-through 4-part continuous forms	Wire matrix line	240 lines/min. @ 10 char/in., 175 lines/min. @ 15 char/in. (assuming 7.4 in. lines)
5225 Printer	1	132	198	4, 6, 8, or 9	3.0- to 17.7-in., 1-through 4-part continuous forms	Wire matrix line	280 lines/min. @ 10 char/in., 195 lines/min. @ 15 char/in. (assuming 7.4 in. lines)
	2	132	198	4, 6, 8, or 9	3.0- to 17.7-in., 1-through 4-part continuous forms	Wire matrix line	400 lines/min. @ 10 char/in., 290 lines/min. @ 15 char/in. (assuming 9.8 in. lines)
	3	132	198	4, 6, 8, or 9	3.0- to 17.7-in., 1-through 4-part continuous forms	Wire matrix line	490 lines/min. @ 10 char/in., 355 lines/min. @ 15 char/in. (assuming 11.8 in. lines)
	4	132	198	4, 6, 8, or 9	3.0- to 17.7-in., 1-through 4-part continuous forms	Wire matrix line	560 lines/min. @ 10 char/in., 420 lines/min. @ 15 char/in. (assuming 13.0 in. lines)
5256 Printer	1	132	---	6 or 8	3- to 15-in., 1-through 4-part continuous forms; 6.0- to 14.5-in. wide individual forms	Serial wire matrix	40 char/sec.
	2	132	---	6 or 8	3- to 15-in., 1-through 4-part continuous forms; 6.0- to 14.5-in. wide individual forms	Serial wire matrix	80 char/sec.
	3	132	---	6 or 8	3- to 15-in., 1-through 4-part continuous forms; 6.0- to 14.5-in. wide individual forms	Serial wire matrix	120 char/sec.

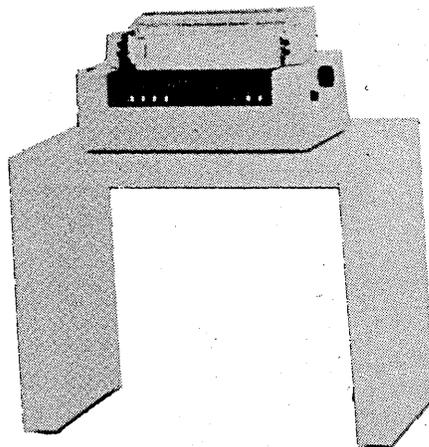
Figure 3-20. 5250 Information Display System Printer Characteristics



IBM 5224 Printer



IBM 5225 Printer



IBM 5256 Printer

Figure 3-21. IBM 5250 Information Display System Printers

Cable Thru Feature

The Cable Thru feature lets you attach multiple display stations or printers to a single cable (Figure 3-22). Each Cable Thru feature provides a port on a work station to which you can connect a twinaxial cable to the next work station. The Cable Thru feature is standard on the 5224; it is a special feature on all other 5250 system devices.

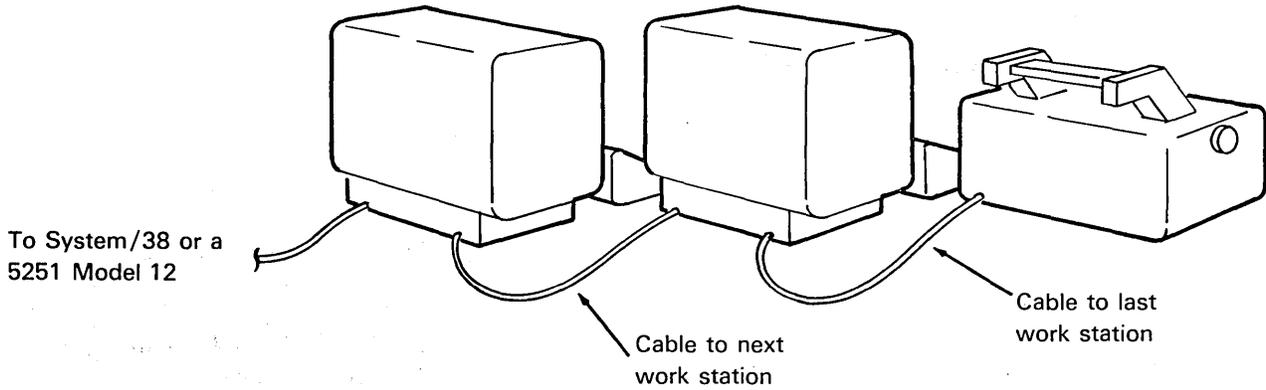


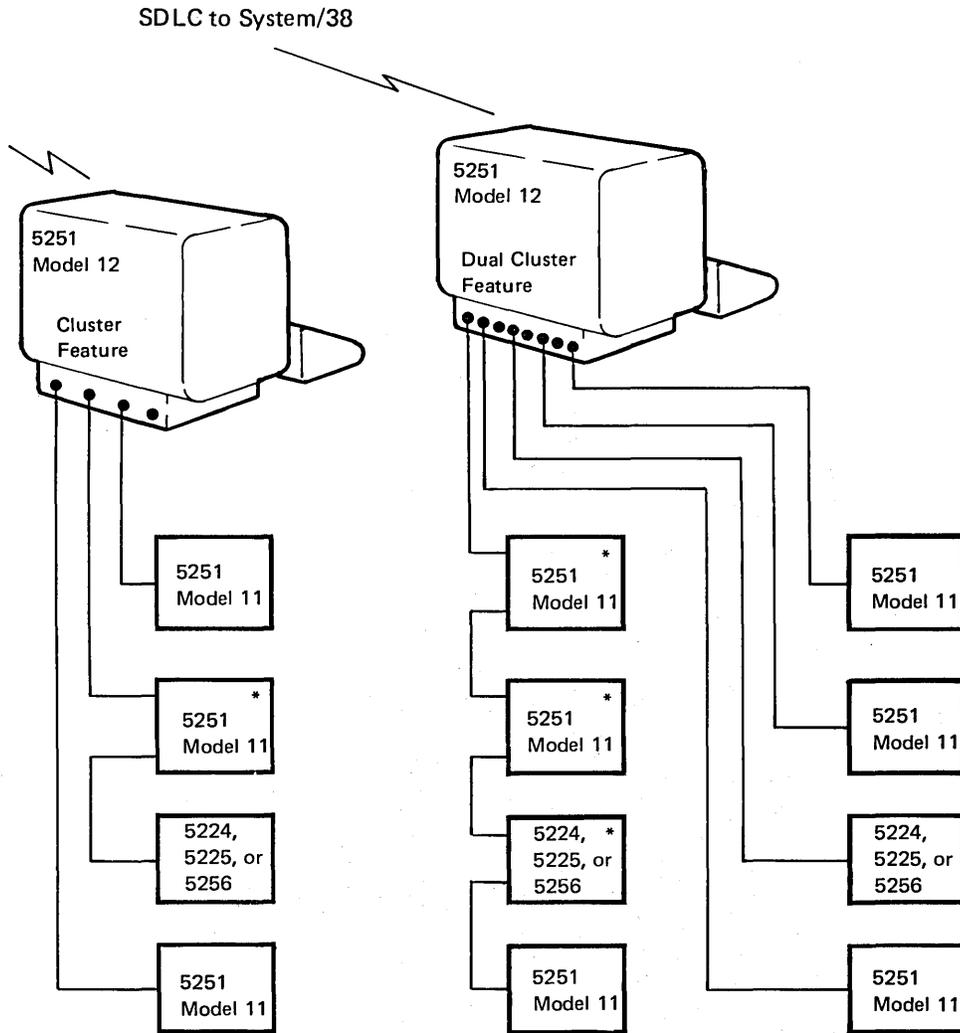
Figure 3-22. Multiple Stations Connected by Cable Thru Features

Cluster Features

Two mutually exclusive cluster features (Cluster or Dual Cluster) are available for the direct attachment of printers and display stations (except the 5251 Model 12) to the 5251 Model 12.

The Cluster feature provides four cable connections, letting you attach up to four work stations. The Dual Cluster feature provides eight cable connections, letting you attach up to eight additional work stations.

Figure 3-23 shows two 5251 Model 12 Display Stations in cluster configurations: one display station has the Cluster feature and the other display station has the Dual Cluster feature. Each 5251 is controlling the maximum number of directly attached work stations.



*Cable Thru feature

Figure 3-23. Cluster Configurations

Keyboards

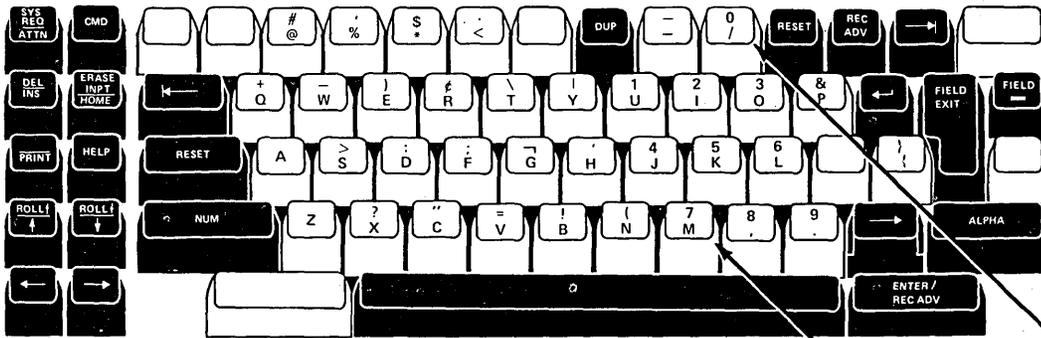
Three different keyboard arrangements are available: the typewriterlike, data-entry, and data-entry with proof arrangement.

The typewriterlike keyboard (Figure 3-24) has the basic keyboard arrangement found on a standard typewriter and an extra set of numeric keys arranged like a calculator keyboard.

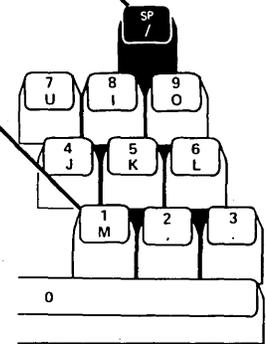


Figure 3-24. 5251 Typewriterlike Keyboard

The data-entry keyboard (Figure 3-25) provides a key arrangement similar to that used on IBM data-entry stations. This keyboard is also available with proof arrangement, an option that features a numeric configuration similar to an adding machine.



Data-Entry Keyboard
(without the Proof
Arrangement feature)



(with the Proof
Arrangement feature)

Figure 3-25. Data-Entry Keyboard

Magnetic Stripe Reader

The Magnetic Stripe Reader feature is supported by System/38 on the 5250 Information Display System. It reads numeric encoded information from a magnetic stripe. The stripe can be used on a wide range of credit cards, operator identification badges, and other magnetic encoded documents. The magnetic stripe can be encoded with as many as 128 ABA (American Banking Association) numeric characters, including control characters. This feature can enhance system data security by requiring a preliminary access to sign-on.

Selector Light Pen

System/38 supports the Selector Light Pen feature on the 5250 Information Display System. The Selector Light Pen is a penlike device that lets you select fields of data from the display screen for system input.

Display Highlighting Functions

Figure 3-26 summarizes the highlighting functions available on a field-by-field basis on the 5251 display screen.

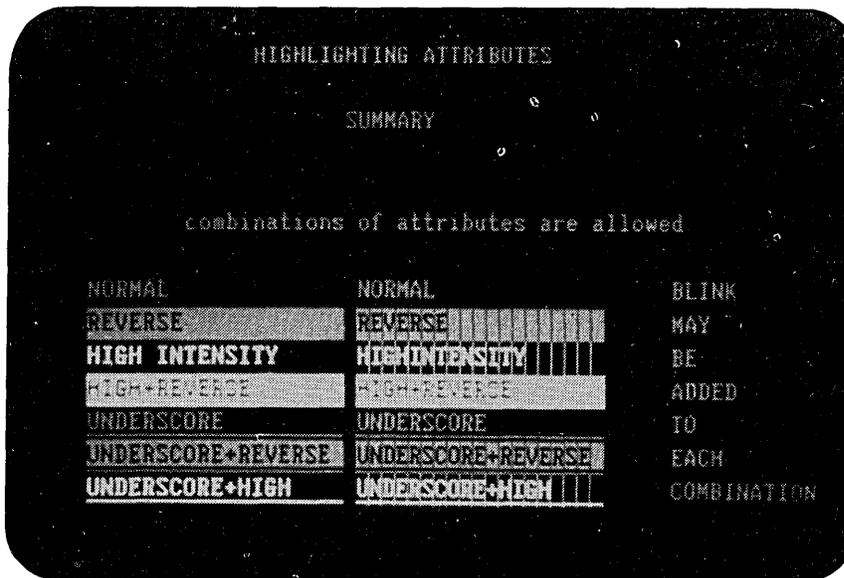


Figure 3-26. Highlighting Function

Blink: Causes all characters in a field to blink repeatedly. The entire field is blinked at the rate of about 1.5 times per second.

Column Separator: Causes all characters in a field to be preceded by and followed by a vertical bar on the 5251. Column separators at the same column position on successive rows appear as a vertical line on the display.

High Intensity: Displays selected characters in a field more brightly than other data on the screen. You can adjust the brightness of high-intensity fields with the contrast control on the display station control panel.

Nondisplay: Lets you enter data without displaying it. This attribute is beneficial in applications where there are security restrictions. By using this system function, you can enter an identification code in a nondisplay field so that the code cannot be seen by others.

Reverse Image: Lets you display the characters in the field as either dark characters on a light background or light characters on a dark background.

You can use the preceding display highlighting methods or a blinking cursor to alert the work station user when an error occurs or when a specified action must be performed.

Underscore: Places a continuous line under all positions of a field.

Chapter 4. Compatibility/Conversion

Even with its major new technological improvements and enhanced functions, System/38 still maintains many commonalities with other IBM products.

IBM System/38

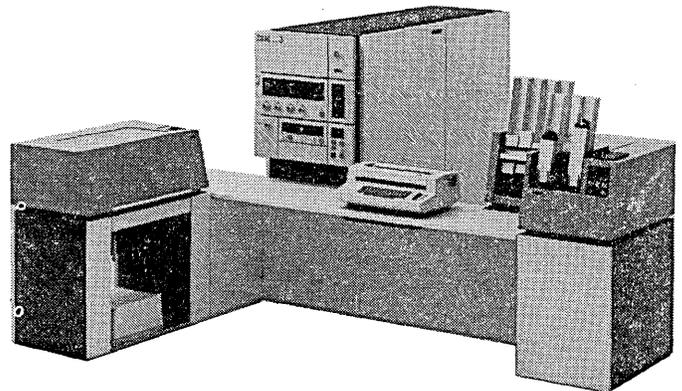
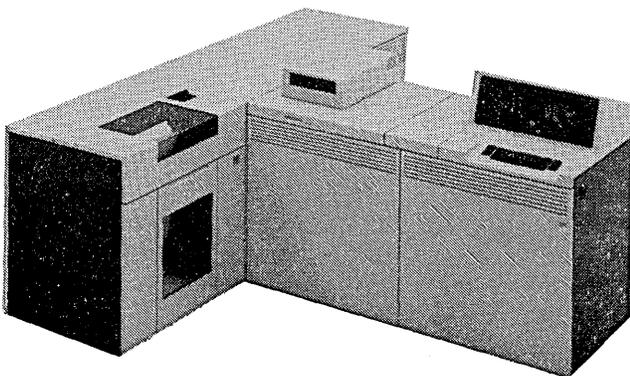
- RPG III
- COBOL
- Control language
- Control language programs
- Diskette support
- 96-column card support
- Magnetic tape support

IBM System/34

- RPG II
- COBOL
- Operation control language
- Procedures
- Diskette support
- No 96-column card support
- No magnetic tape support

IBM System/3

- RPG II
- COBOL
- Operation control language
- Procedures
- No diskette support
- 96-column card support
- Magnetic tape support



COMPATIBILITY WITH SYSTEM/3 AND SYSTEM/34

Even with the major new technological improvement it represents, System/38 can still appear very much like System/3 or System/34 to the programmer, if required. He writes his application in RPG III or COBOL using similar file structures.

With certain required source changes and compilation, existing System/34 and System/3 batch programs execute on System/38. Data files can be moved from System/3 to System/38. This compatibility lets you move working application programs to System/38. The programs can immediately use data base data management functions. Also, you can develop new applications that use this data in the data base.

System/38 includes most of the utility functions used in System/34 and System/3, but the need for some functions and their effects are different.

The System/38 RPG III language is an enhanced extension of the RPG II language. System/38 data base data management function offers all the access methods available on the System/34 and System/3 Models 4, 8, 10 Disk, 12, or 15. The described data enforces field length and type definitions that were previously specified only in RPG II programs. The data field names are directly available to the System/38 RPG III compiler, without the need for redefinition in each program. You simply define a field to the system once; thereafter, using that field in a program simply requires specifying the field name.

System/38 communications facilities for the 5250 devices use SDLC (synchronous data link control).

System/38 also supports reading and writing basic data exchange diskettes that can be used on System/34 and System/3. System/38 also supports the reading and punching of 96-column cards, and the reading and writing of magnetic tape.

CONVERSION FROM SYSTEM/3 BATCH

A System/38 conversion utility licensed program is available to help the System/3 user convert to System/38 (this program executes on System/3).

The IBM System/38 System/3 Batch Conversion Utilities is a one-time-charge licensed program that automates the actual details of conversion while letting you control the overall conversion process. The utilities run on System/3 Models 8, 10 Disk, 12, and 15 and will convert batch systems for System/3 Models 4, 8, 10 Disk, 12, and 15. The utilities let you do the major part of your conversion on the installed System/3 before your System/38 arrives.

The System/3 Batch Conversion Utilities convert most programming elements of a System/3 installation to their System/38 equivalents. This includes:

- Converting System/3 RPG II source programs to System/38 RPG III source programs
- Converting System/3 RPG II auto report source programs to System/38 RPG III auto report source programs
- Converting System/3 procedures to System/38 control language programs
- Converting System/3 operational control language to System/38 control language
- Converting selected System/3 utility statements to their System/38 equivalents
- Generating data descriptions for System/3 disk files that are to be moved to the System/38 data base

You control the actual conversion process. You can elect to let the utility automatically convert, or you can intervene and selectively choose what operational control language and applications are to participate in the conversion process.

For corrections necessary during conversion, and because you may want to alter the conversion output, the System/38 Batch Conversion Utilities include a maintenance service package. By using maintenance service, you can resolve and correct statements that need special attention or modify the conversion process so that the output meets your particular needs.

Throughout the conversion process, communication is maintained so that you will be able to understand what is happening. In addition, output procedures are provided so that you can print or punch any of the elements that are participating in the conversion.

Besides performing the actual conversion, the Batch Conversion Utilities generate the interchange media that will be used to move the converted material from System/3 to System/38. Again, you control what interchange media fits your system and whether you will convert your system in steps or will do the whole conversion as a single operation.

Figure 4-1 shows an overview of the Batch Conversion Utilities.

For applications developed on System/38, a sort utility is not required because different sequences of data records are obtained by creating logical files that specify the required access paths. Almost all functions achieved with the System/3 Sort Utility can be accomplished with logical files. The exceptions are sort FORCE (which performs data substitution as the data is sorted) and summary sort (which summarizes the data as it is sorted).

To simplify System/3 sort to System/38 sort conversion, a Conversion Reformat Utility licensed program (which executes on System/38) is available. This utility creates either a record address file or a reformatted physical file. The Conversion Reformat Utility duplicates all functions of the System/3 sort except summary sort. In addition, the Conversion Reformat Utility uses System/3 sort specifications.

The input to the conversion is the user's source libraries, operation control language (OCL), job streams, data files, and procedures.

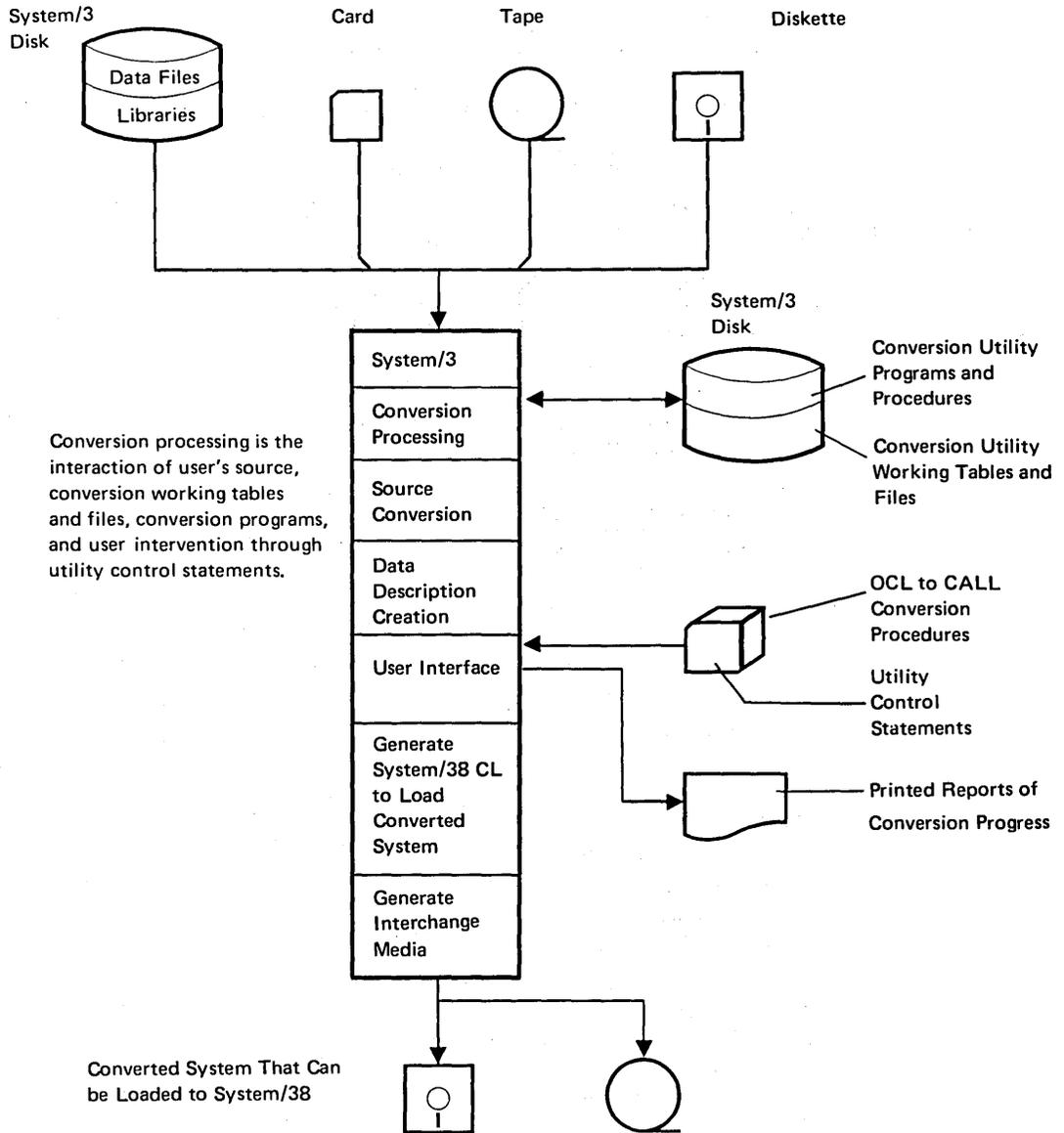


Figure 4-1. System/3 Batch Conversion Utilities Overview

Programs

System/38 RPG III language is source-level compatible with the System/3 RPG II language. Some device-dependent code, however, will have to be changed (for example, DISK45 and DISK40). Some RPG II statements (such as exits to System/3 Assembler routines) will also be flagged to show the need for user action.

Utilities

The Conversion Utility converts selected System/3 utilities (such as copy and delete) to System/38 control language commands. The Conversion Utility flags those System/3 utilities not converted (such as file maintenance). Most System/3 converted utility functions are supported with easy-to-use control language functions.

Operational Control Language

The System/3 Batch Conversion Utilities convert the System/3 operational control language (OCL) to the System/38 control language equivalent.

Data

Data, RPG II, and control language source must be moved to System/38 from System/3. It is not necessary to reorganize data files because System/38 supports the data organizations used on System/3.

The System/3 Batch Conversion Utilities build external data descriptions based upon System/3 Volume Table of Contents (VTOC) information, the RPG file description, and input specifications. The System/3 Batch Conversion Utilities then let you optionally expand and modify the data descriptions provided by the utilities, as well as let you create additional descriptions. In addition, the utility provides cross-reference reports.

The System/3 Batch Conversion Utilities build a data stream that is written to diskette or magnetic tape and consists of:

- Converted programs with control language generated for compiling them
- Control language from the operational control language
- Data descriptions for your data files with control language for creating them
- Control language to load the data from System/3 into the data files

CONVERSION FROM SYSTEM/3 CCP

The IBM System/38 CCP (communications control program) Conversion Aid licensed program helps the System/3 user convert to System/38. This aid is directed to CCP systems that use RPG II programs and 3270 devices attached locally or remotely. This aid handles only those 3270 devices that are programmed through the Display Format Facility (DFF).

This aid operates on a System/3 Model 8, 10 Disk, 12, or 15 under control of the current version of the SCP (system control program) appropriate to the System/3 model. The minimum System/3 requirements are:

- 40 K bytes of user program main storage
- 5 million bytes of disk storage

During the conversion process, the conversion aid:

- Converts DFF specifications for display formats into DDS (data description specifications) for System/38 display files. DFF specifications for printer formats are not converted.
- Converts RPG II specifications into RPG III specifications.
- Generates record level DDS for System/38 data base files (if you request data base DDS) and generates RPG III mainline programs.
- Converts sort specifications into Reformat Utility specifications, and generates System/38 control language to run the converted sort program.
- Provides notification where your programming techniques need to be clarified and where partial rewrite or redesign is required.

CONVERSION FROM SYSTEM/34

The IBM System/34 to System/38 Conversion Aid helps the System/34 user convert to System/38. The conversion program is menu-oriented; the user selects tasks on the menu for which the conversion program generates jobs to be done in a batch mode.

The conversion program provides the following functions:

- Prints converted code listings. Statements that cannot be converted and that are found in the conversion aid are flagged.
- Prints conversion control log on demand for auditing purposes.
- Converts the following:

System/34	System/38
RPG II	RPG III
Operational control language	Control language
Display format specifications	Data description specifications
Sorts	Reformat utility
Menus	Data description specifications, control language
Message member	Message file
Data files	DDS and data files

- Generates the following System/34 functions on System/38 through programming:
 - RPG II SUBR21 – Access to local library area (LDA)
 - RPG II SUBR23 – Message member retrieval (single level)
- Unloads converted source code from the System/34 to diskette in standard data interchange format.
- Loads standard data interchange format diskettes containing the converted source code to System/38.
- Provides for parameter substitution into converted sort specifications on System/38.
- Provides a menu for compiling on System/38.

access path: The means by which CPF provides a logical organization to the data in a data base file so that the data can be processed by a program. See also *arrival sequence access path* and *keyed sequence access path*.

activity level: An attribute of a storage pool or the system that specifies the maximum number of jobs that can execute concurrently in the storage pool or the system.

arrival sequence access path: An access path that is based on the order in which records are stored in a physical file.

authority: The right to access objects, resources, or functions.

autostart job: A job that is automatically initiated when a subsystem is started.

auxiliary storage: All addressable storage other than main storage. Auxiliary storage is located on the system's nonremovable disk enclosures.

batch job: A group of processing actions submitted as a predefined series of actions to be performed without a dialog between the user and the system.

batch subsystem: A subsystem in which batch jobs are to be processed. IBM supplies one batch subsystem: QBATCH.

breakpoint: A place in a program (specified by a command or a condition) where the system halts execution and gives control to the work station user or to a specified program.

class: An object that contains the execution parameters for a routing step. The system recognized identifier is *CLS.

command: A statement used to request a function of the system. A command consists of the command name, which identifies the requested function, and parameters.

control language: The set of all commands with which a user requests functions. Abbreviated CL.

control language program: An executable object that is created from source consisting entirely of control language commands.

control language variable: A program variable that is declared in a control language program and is available only to the program.

Control Program Facility: The system support licensed program for the IBM System/38. It provides many functions that are fully integrated in the system such as work management, data base data management, job control, message handling, security, programming aids, and service. Abbreviated CPF.

control storage: High-speed storage containing microcode.

controlling subsystem: An interactive subsystem that is started automatically when the system is started and through which the system operator controls the system. IBM supplies one controlling subsystem: QCTL.

CPF: See *control program facility*.

data area: An object that is used to communicate data such as CL variable values between the programs within a job and between jobs. The system-recognized identifier is *DTAARA.

data base: The collection of all data files stored in the system.

data base file: Of related records in the data base. See *physical file* and *logical file*.

data description specifications (DDS): A description of the user's data base or device files that is entered using a fixed-form syntax. The description is then used to create files.

data file utility (DFU): The utility of the Interactive Data Base Utilities licensed program that is used to create, maintain, and display records in a data base file.

data rights: The authority to read, add, update (modify), or delete data contained in an object.

DDS: See *data description specifications*.

device description: An object that contains information describing a particular device that is attached to the system. The system-recognized identifier is *DEVD.

device file: An object that describes the data processed on an external input or output device attached to the system, such as a work station, a card device, a printer, the diskette magazine drive, or magnetic tape drive.

DFU: See *data file utility*.

exception: An abnormal machine condition that is detected during the execution of a particular function.

externally described data: Data contained in a file for which the fields in the records are described to CPF, by using data description specifications, when the file is created. The field descriptions can be used by the CL, RPG, or COBOL program when the file is processed.

field: An area that is reserved and used for a particular item of information.

field reference file: A physical file whose record format describes the fields used by a group of files but which contains no members. The field descriptions in the field reference file can be referred to when data description specifications for other files are written. Therefore, data attributes only need to be specified once for fields used in multiple files.

file: An object that contains a description of a set of related records treated as a unit and, optionally, those records. The system-recognized identifier is *FILE.

file description: The information contained in the file that describes the file and its contents.

file overrides: The file attributes specified at execution time that will override the attributes specified in the file description or in the program.

first-level message: The initial message presented to the user containing general information or designating an error.

general purpose library (QGPL): The library provided by CPF to contain user-oriented, IBM-provided objects and user-created objects that are not explicitly placed in a different library when they are created.

help text: Information associated with a menu, message, or prompt that explains options or values displayed on the menu or prompt, or provides more information about the message. Help text is requested by pressing the Help function control key.

inline data file: A data file that is included as part of a job when the job is read from an input device by a reader program.

input stream: A group of records submitted to the system as batch input that contains CL commands for one or more jobs and/or the data records of one or more inline data files.

inquiry: A request for information from a data file usually made against one record.

integrity: The protection of data and programs from inadvertent destruction or alteration.

interactive: Pertaining to a program or system that alternately accepts input and then responds. An interactive system is conversational, that is, a continuous dialog exists between the user and the system.

Interactive Data Base Utilities (IDU): A System/38 licensed program that consists of DFU, SEU, SDA, and Query.

interactive job: A job in which the processing actions are performed in response to input provided by a work station user. During a job, a dialog exists between the user and the system.

interactive subsystem: A subsystem in which interactive jobs are to be processed. IBM supplies three interactive subsystems: QCTL, QINTER, and QPGMR.

job: A single identifiable sequence of processing actions that represents a single use of the system. A job is the basic unit by which work is identified on the system.

job description: An object that contains the attributes of a job. The system-recognized identifier is *JOBID.

job log: A record of requests submitted to the system by a job and the messages related to them. The job log is maintained by CPF.

job queue: An object on which batch jobs are placed when they are submitted to the system and from which they are selected for execution by CPF.

key field: A field of a record whose contents are used to sequence the records of a particular type within the file.

keyed sequence access path: An access path to a data base file that is based on the contents of key fields contained in the individual records.

library: An object that serves as a directory to other objects. A library is used to group related objects and to find objects by name when they are used. The system-recognized identifier is *LIB.

library list (*LIBL): An ordered list of library names used to find an object. The library list indicates which libraries are to be searched and the order in which they are to be searched. *LIBL specifies to the system that a job's current library list is to be used to find the object.

local work station: A work station that is connected directly to a System/38 without need for data transmission facilities. Contrast with *remote work station*.

logical file: A data base file through which data that is stored in one or more physical files can be accessed by means of record formats and/or access paths that are different from the physical representation of the data in the data base.

main storage: All storage in a computer from which instructions can be executed directly.

member: An identifiable group of records that is a subset of the data base file to which it belongs. Each member conforms to the characteristics of the file and has its own access path.

menu: A type of display in which a list of options is shown.

message: A communication sent from one person or program to another person or program.

message description: The descriptive information about a message and the text of the message.

message queue: An object in which messages are placed when they are sent to the person or program. The system-recognized identifier is *MSGQ.

modem: A mechanism that modulates and demodulates signals transmitted over data communication facilities.

object: A named unit that consists of a set of attributes (that describe the object) and data. An object is anything that exists in and occupies space in storage and on which operations can be performed. Some examples of objects are programs, files, and libraries.

object rights: The authority that controls what a system user can do to an entire object. For example, object rights can let a user delete, move, or rename an object.

output queue: An object that contains a list of output files to be written to an output device by a writer. The system-recognized identifier is *OUTQ.

password: A unique string of characters that a system user enters to identify himself to the system.

physical file: A data base file that contains data records. All the records have the same format; that is, they are all fixed-length records and they all contain the same fields in the same order.

problem determination: The process of determining the source of a problem as a component problem, a machine failure, a common carrier link, a user-supplied element, or a user error.

program described data: Data contained in a file for which the fields in the records are described in the program that processes the file.

programmer subsystem: An interactive subsystem in which programmers can perform online programming through 5251 and 5252 display stations. IBM supplies a programmer subsystem: QPGMR.

prompt: A request for information or user action. The user must respond to allow the program to proceed.

qualified object name: An object name and the name of the library containing the object.

query: A request to extract, from a file, one or more records based upon a criterion.

reader: A program that reads jobs from an input device or data base file and places them on a job queue.

record: An ordered set of fields that make up a single occurrence of a record format. A record is the basic unit of data transferred between a file and a program.

record format: The definition of how data is structured in the records contained in a file. The definition includes the record name and field descriptions for the fields contained in the record. The record formats used in a file are contained in the file's description.

remote work station: A work station whose connection to the processing system uses modems and common carrier or private data transmission facilities. Contrast with *local work station*.

routing data: A character string that CPF compares with character strings in the subsystem description routing entries to select the routing entry that is to be used to initiate a routing step. Routing data can be provided by a work station user, specified in a command, or provided through the work entry for the job.

routing entry: An entry in a subsystem description that specifies the program to be invoked to control a routing step that executes in the subsystem.

routing step: The processing performed as a result of invoking a program specified in a routing entry.

Screen Design Aid: This aid helps the programmer to design screen formats for System/38 work stations without entering all the code used for formatting.

SDA: See *screen design aid*.

second-level message: Provides additional information to that already provided in a first-level message. Second-level messages are obtained by pressing the Help key while a first-level message is displayed.

sector: A portion of a diskette track.

security: The control of access to, or use of, data or functions.

SEU: See *source entry utility*.

shared access path: An access path used by more than one file to provide access to data common to the files.

Source Entry Utility: This aid lets the programmer enter or modify source statements interactively from any work station.

source file: A file created to contain source statements for such items as high-level language programs and data description specifications.

spooled file: A device file that provides access to data processed by readers or causes output data to be saved for later processing by a writer.

storage pool: A logical segment of main storage reserved for executing a group of jobs.

subsystem: A predefined operating environment through which CPF coordinates work flow and resource usage.

subsystem attributes: Specifications in a subsystem description that specify the amount of main storage available to the subsystem and the number of jobs that can execute concurrently in the subsystem.

subsystem description: An object that contains the specifications that define a subsystem and that CPF uses to control the subsystem. The system-recognized identifier is *SBSD.

system library: The library provided by CPF to contain system-oriented objects provided as part of CPF. Named QSYS.

temporary library: A library that is automatically created for each job to contain temporary objects that are created by that job. The library is deleted when the job ends. The system-recognized identifier is QTEMP.

user identification: Recognition of a system user so that only the facilities and data he is authorized to use are made available to him.

user profile: An object that contains a description of a particular user or group of users. A user profile contains a list of authorizations to objects and functions. The system-recognized identifier is *USRPRF.

variable: A named modifiable value. The value can be accessed or modified by referring to the name of the variable.

virtual storage: The combination of main storage and auxiliary storage, treated as a single addressable unit.

work entry: An entry in a subsystem description that specifies a source from which jobs can be accepted to be executed in the subsystem.

work station: A device that lets a person transmit information to or receive information from a computer, or both, as needed to perform his job.

writer: A CPF program that writes spooled output files from an output queue to an external device, such as a printer.

Index

- application development 1-6
- audible alarm and attention light 3-8
- auto report, RPG III 2-46
- automatic initial microprogram load 3-7
- auxiliary storage 3-13

- batch conversion, System/3 1-10, 4-2, 4-6
- batch input streams 2-12
- binary synchronous communication 3-30

- cable thru feature 3-37
- cables, work station controller,
 - twinaxial 3-27
- card device 3-23
- channel, I/O 3-6
- clock, time of day 3-7
- cluster feature 3-38
- COBOL 1-8, 2-47
 - extensions supporting data base and work station processing 2-49
 - programmer services 2-50
- command entry interface 1-4
- command grouping menu 2-10
- commands (see *Control Language*) 2-9
- communications 3-25
- communications attachment 3-29
- communications control program (CCP)
 - conversion aid 1-10, 4-6
- compatibility, with System/3 and System/34 4-1
- compilation listing functions, RPG III 2-45
- concurrent functions 1-4
- configuration viii, 3-2
- console, system 3-8
- control adapter, system 3-12
- control language 1-4, 2-9
- control language programs 2-11
- control program facility (CPF) 2-2, 1-3
- control program facility functions 2-3
- conversion from System/3 batch 4-2
- conversion from System/3 CCP 4-6
- conversion from System/34 4-7
- conversion reformat utility v, 1-10, 2-57
- copy 2-34

- data base data management 2-22
- data base file processing 2-26
- data base files 2-23

- data, conversion of 4-5
- data description specifications (DDS) 2-19
- data, externally described 1-5, 2-19
- data file utility 2-53
- data management
 - card device 2-30
 - data base 2-22
 - device 2-27
 - diskette device 2-30
 - display device 2-28
 - printer device 2-30
 - work station 1-4
- DDS (see data description specifications)
- debug facilities, RPG III 2-45
- debugging 2-32
- detection, power failure 3-8
- device configuration 2-37
- device control expansion 3-27
- device data management 2-27
- device interface expansion 3-28
- direct access storage device 3-14
- disk enclosures 3-13
- diskette 3-16
- diskette magazine drive 3-15
- display device data management 2-28
- display highlighting functions, 5250 3-41
- display screen, system console 3-9
- display station 3-31
- displays 2-4

- electromagnetic interference iii
- externally described data 1-5, 2-19

- failure, AC power, detection of 3-8
- FCC regulations iii
- file processing 2-26
- file reference 2-33
- files
 - logical 2-23
 - physical 2-23

- high-level machine interface 1-2, 3-3
- highlighting, display 3-41
- history log 2-36

- information display station 3-32
- input/output channel, internal 3-6
- input/output devices 1-13
- input streams, batch 2-12
- integrated modem 3-29
- interactive data base utilities 1-9, 2-52
 - data file 1-9, 2-52
 - query 1-9, 2-53
 - screen design aid 1-9, 2-53
 - source entry 1-9, 2-51
- interfaces, user and machine 1-2, 2-4
- internal input/output channel 3-6
- initial microprogram load, automatic 3-7
- inquiry 2-54

- job log 2-36
- job streams, batch 2-12
- jobs 2-14

- keyboards 3-39
 - data entry, 5250 3-40
 - system console 3-8
 - typewriter-like 3-39
- keylock, power 3-7

- language enhancements, RPG III 2-41
- libraries 2-13
- log
 - history 2-36
 - job 2-36
- logical files 2-23

- machine 1-12
- machine configurator 3-2
- machine interface, high-level 1-2, 3-3
- machine summary 3-1
- magazine, diskette 3-15
- magnetic stripe reader, 5250 3-41
- magnetic tape subsystem 3-24
- main storage capacity 3-7
- managing work 2-14
- message handling 1-6, 2-34
- messages 2-36
- MFCU 3-23
- microprogram load, automatic initial 3-7
- modem, integrated 3-29
- multifunctional card unit 3-23

- object management 2-12
- objects 2-12
- online program development 1-5
- operator/service panel, system console 3-11

- physical files 2-23
- power failure detect 3-8
- power keylock 3-7
- power warning feature 3-8
- print belts 3-19
- print cartridge 3-19
- printers 3-17
 - 3203 3-22
 - 3262 3-21
 - 5211 3-20
 - 5224 3-34
 - 5225 3-34
 - 5256 3-34
- processing unit 3-4
- program products 1-3, 2-1
- programmer services 2-31
- programs, conversion 4-6, 4-7

- query utility 2-54

- remote communications 3-29
- RPG III 1-7, 2-41
 - auto report 2-46
 - compilation listing functions 2-45
 - debug facilities 2-46
 - enhancements 2-41

- save/restore 2-39
- screen design aid (SDA) utility 2-53
- screen, display 3-9
- SDA (screen design aid) 2-53
- security 1-5, 2-37
- selector light pen, 5250 3-41
- service 1-6, 2-39
- service panel, system console 3-11
- services
 - programmer 2-31
 - system 2-36
 - system operator 2-35
- source entry utility 2-52
- spooling 1-5, 2-16
- storage management 3-6
- subsystems 2-15
- synchronous data link control (SDLC)
- support 3-29

- system configuration viii, 3-2
- system console
 - display screen 3-9
 - keyboard 3-10
 - operator/service panel 3-11
- system control adapter 3-12
- system operator services 2-35
- system services 2-36
- system unit 1-12, 3-3
- System/3 batch conversion
 - utilities 1-10, 4-2
- System/3 and System/34 compatibility 4-1
- System/34 conversion aid utility 4-7

- test environment 2-32
- testing 2-32
- throughput with spooling 2-17
- time-of-day clock 3-7
- twinaxial cable 3-27

- user interfaces 2-4
- utilities, conversion 4-6
- utilities, System/38 2-52

- virtual address translator 3-6

- work station controllers 3-26
 - basic 3-26
 - device control expansion 3-27
 - device interface expansion 3-28
 - second, third, and fourth 3-28
 - twinaxial cable 3-27
- work stations 3-32
- work, managing 1-5, 2-14

- 3203 printer 3-17, 3-22
- 3262 printer 3-17, 3-21
- 3370 direct access storage device 3-14
- 3410/3411 magnetic tape subsystem 3-24
- 5211 printer 3-17, 3-20
- 5224 printer 3-34
- 5225 printer 3-17, 3-34
- 5250 information display system 3-32
- 5251 display station 3-33
- 5256 printer 3-34
- 5381 system unit 3-3
- 5424 multi-function card unit 3-23

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your nearest IBM branch office.

If your comment does not need a reply (for example, pointing out a typing error) check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s):

Comment(s):

Please contact your nearest IBM branch office to request additional publications.

Name _____

Company or
Organization _____

Address _____

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

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

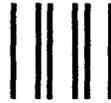
City State Zip Code

Cut Along Line

Fold and tape

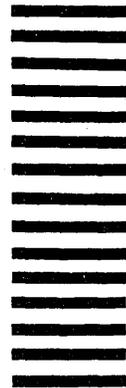
Please do not staple

Fold and tape



NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N. Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

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

Fold and tape

Please do not staple

Fold and tape



International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30055
(U.S.A. only)

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

IBM System/38 Introduction (File No. S38-00) Printed in U.S.A. GC21-7728-4

