GC21-7728-0

File No. S38-00

IBM System/38

IBM System/38 Introduction

Preface

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

The purpose of this publication is to help the reader achieve a general understanding of this new general purpose system and the capabilities it gives the user.

This introduction presents:

- A summary of the System/38 design and highlights of its major functions.
- A fundamental description of data processing concepts in terms of the System/38 program products and machine.
- A description of the system configurations and the machine characteristics.
- A description of the compatibility that exists between System/38 and System/3.

Note: This publication follows the convention that *he* means *he or she*. Also, the reader is expected to have a basic knowledge of data processing.

Related Publications

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

First Edition (October 1978)

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

This publication is for planning purposes only. The information herein is subject to change before the products described become available.

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

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

Contents

CHAPTER 1. INTRODUCTION TO IBM SYSTEM/38 . 1-1
Program Products 1-3
  Control Program Facility (CPF) 1-3
  RPG III 1-6
  Interactive Data Base Utilities (IDU) 1-7
  System/3 Batch Conversion Utilities 1-7
  Conversion Reformat Utility 1-8
Machine 1-8
  System Unit 1-8

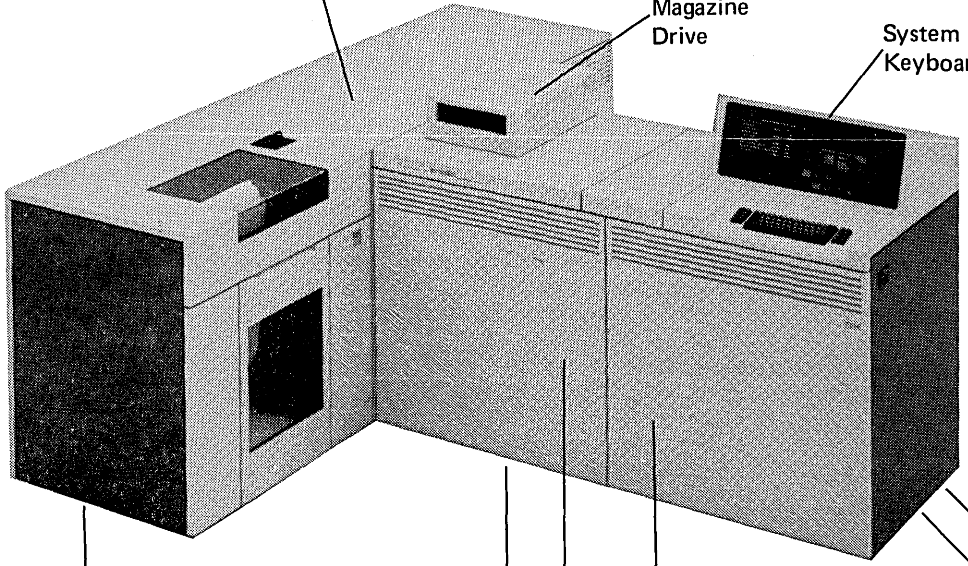
CHAPTER 2. SYSTEM/38 PROGRAM PRODUCTS . . 2-1
Control Program Facility (CPF) 2-2
  User Interfaces 2-4
  Control Language Programs 2-9
  Batch Job Streams 2-10
  Object Management 2-10
  Managing Work 2-12
  Data Description Specifications (DDS) 2-15
  Data Base Data Management 2-18
  Device Data Management 2-22
  Programmer Services 2-25
  Testing and Debugging 2-26
  System Operator Services 2-29
  System Services 2-30
  RPG III 2-33
Interactive Data Base Utilities 2-39
  Source Entry Utility (SEU) 2-39
  Data File Utility (DFU) 2-40
  Query Utility 2-41
Conversion Reformat Utility 2-41

CHAPTER 3. MACHINE SUMMARY 3-1
System Unit 3-3
  High-Level Machine Interface 3-3
  Processing Unit 3-4
  Storage Management 3-5
  Virtual Address Translator 3-5
  Internal Input/Output Channel 3-5
  Main Storage Capacity 3-6
  Time-of-Day Clock 3-6
  Automatic Initial Microprogram Load 3-6
  Power Keylock 3-6
System Console 3-7
  System Console Display Screen 3-8
  System Console Keyboard 3-9
  System Console Operator/Service Panel 3-10
Auxiliary Storage 3-11
Diskette Magazine Drive 3-12
Printers 3-15
  5211 Printer; 3262 Printer 3-15
Card Device 3-19
  5424 Multi-Function Card Unit 3-19
Magnetic Tape Device 3-20
  3410/3411 Magnetic Tape Subsystem 3-20
Communications Adapters 3-21
  Work Station Controllers 3-22
  Communication Attachment 3-25
Work Stations 3-26
  5250 Information Display System 3-26

CHAPTER 4. COMPATIBILITY/CONVERSION 4-1
Compatibility with System/3 4-1
Conversion from System/3 4-2

GLOSSARY G-1

INDEX X-1



Expansion Enclosure

- Auxiliary Storage (64.5–258MB)

Diskette Magazine Drive

System Console Keyboard/Display Screen

Printer

- 5211 (Prints up to 300 lpm)
- 3262 (Prints up to 650 lpm)

Communications

- Communication Attachment (remote)
  - Integrated Modems
  - DDSA
  - EIA/CCITT
  - TP Support
    (SDLC)
    (HDX Primary)
    (HDX Secondary to S/370)
- Work Station Controller (local)
  (up to 40 work stations)

Base Enclosure

- 5381 Processing Unit
- Main Storage
  512–1024K (Model 3)
  512–1536K (Model 5)
- Auxiliary Storage
  (64.5–129MB)

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)

5250 Information Display System

Chapter 1. Introduction to IBM System/38

IBM System/38 is a general-purpose, data processing system designed to significantly improve the user's productivity in developing, maintaining, and enhancing applications for the interactive work station environment.

The system provides 512K to 1536K bytes of main storage, 64.5 to 387 megabytes of nonremovable auxiliary (disk) storage, a diskette magazine drive, and a system console with keyboard and display. A multi-function 96-column card reader/punch and up to two 650-line per minute system printers may be attached. The system supports direct attachment of up to 40 local 5250 work stations (displays and printers) and a large number of remotely attached work stations through SDLC teleprocessing lines.

The work stations can be placed where they are needed in the organization (in the offices, departments, plants, and the warehouses) so that company personnel can share a common data base and the processing power of a computer. System/38 supports this work station environment with the function required to maintain up-to-date business data available for retrieval on request by any of the work station users. Providing these users with information that is current and accurate can greatly improve the efficiency of an organization.

The work station environment requires many unique functions, such as the ability for work station users to share programs, data files, and system resources without significant delays to any user. This environment also requires the ability to access data records by more than one access path (index). For example, one user may need to access inventory item records by item stock number while another user is accessing these same records by vendor number.

On most current systems, the unique functions required by the work station environment have either been added onto the existing batch programming system or they are programmed by the user in the application programs. In contrast, System/38 is an entirely new system developed specifically for the work station environment. The needed function has been designed into the system. Major advances in the level of system function, and the ease of use of that function, enable the System/38 user to develop work station applications in a shorter time and with a lower level of expertise than was required on previous systems.

System/38 can support a range of environments from one consisting almost entirely of batch processing to one that makes extensive use of work station applications. System/38 is designed to allow the user to make easy 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 has been designed to manage a mixture of batch and interactive work providing fast response to work station user requests and good throughput for batch jobs.

In developing System/38, first the desired functions were determined. It was then determined which parts of these functions should be performed by the control program and which parts should be performed by the *machine.* Next a machine was designed with a functionally high level interface (Figure 1-1). The machine does much more than just add, subtract, move data, and execute I/O. The machine performs complete functions such as: finding a file or program in system storage, locating a record in a file through a key (index), initiating a new process, manipulating queues, and allocating storage.

The high-level System/38 machine interface ensures compatibility of the Control Program Facility and the user's programs implemented on different models of the machine. New physical devices can be added to the system without requiring reprogramming. The user can expect greater reliability because basic system functions are built into the *machine.*

Additional high-usage functions can be implemented in the circuitry of the machine to provide improved performance without impacting the Control Program Facility or the user's programs. New functions can be added to the machine interface because of its extendable design.

IBM's emphasis on advanced technology and the development of a highly integrated system enables System/38 to provide users with an improved level of productivity, ease of use, data integrity, functional capability, reliability, and non-disruptive growth in applications.

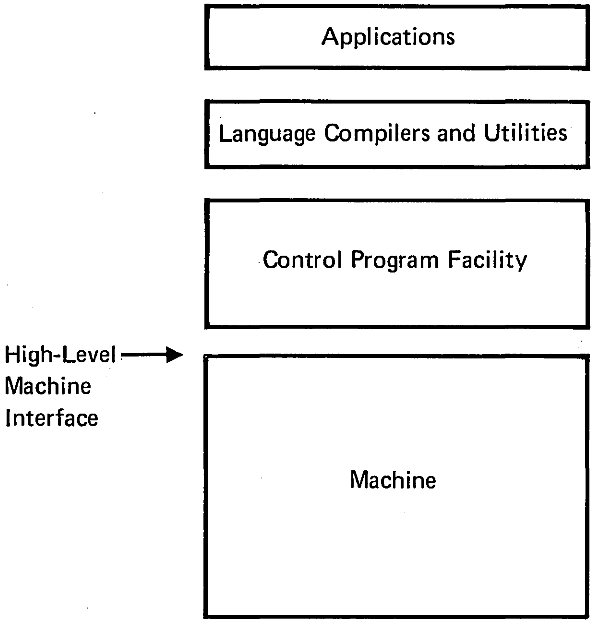

Applications
Language Compilers and Utilities
Control Program Facility
High-Level Machine Interface
Machine

Figure 1-1. User and Machine Interfaces

PROGRAM PRODUCTS

The program products developed for System/38 include:

- Control Program Facility
- RPG III
- Interactive Data Base Utilities
- System/3 Batch Conversion Utilities
- Conversion Reformat Utility

Control Program Facility (CPF)

The Control Program Facility is the system support program product for System/38. It provides many integrated functions that are designed to satisfy the total installation requirements for a multiprogrammed, batch, and online interactive system.

Some of the major new functions that greatly increase the ease of use and application productivity are:

Multiple immediately maintained access paths shared concurrently by multiple users and programs performing read, update, add, and delete record operations. These functions are directly supported in RPG III. The Control Program Facility also provides RPG III with the ability to 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). This support eliminates the 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 user programs can be designed and coded as if there will only be a single user of the program at a time. There are no specifications or programming considerations required for multiple users. However, any number of work station users may be executing the program at the same time. The number of users never needs to be specified. This 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 there were multiple copies.

Work station data management functions supported directly in RPG III allow the user to define display screen formats outside the program and easily perform interactive functions with a minimal amount of coding.

A control language (CL) that considerably simplifies the user interface to the system while providing extensive new function. The control language is made up of commands that have consistent, easy-to-learn names and syntax. The control language prompting support simplifies the specification of command parameters.

In addition to specifying the system functions, the control language provides commands for logic control, use of variables, data manipulation, work station input and output, and arithmetic functions. Therefore, the control language is a high-level, user programming language for system functions and control language programs can be written and compiled for efficient performance.

A command entry interface that allows control language commands to be entered interactively from any work station. Ease of use is provided by prompting for command parameters and providing defaults for most parameters.

A multi-user system where all of the system functions are available at all work stations through the control language. The system allows the work station users to perform system functions concurrently as if each were the only system user. For example, several programmers can all be entering source and 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 allows the programmer to describe each data field in the data base. On previous systems the data fields were described in the application programs. With external data description, 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 changes required to programs when changes are made to the data base.

Online program development functions that allow the programmer to easily enter source statements, compile programs, view the results of compilation, correct source statements, and test the execution of programs from any work station. Interactive debug facilities allow the programmer to find and correct execution errors in a program in terms of the high-level language. All of these activities can be carried on concurrently with normal production work on the system.

Work management functions that simplify the user actions necessary to control all forms of work on the system, ranging from a simple command execution to the scheduling and execution of batch jobs and to the control of interactive jobs 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 high efficiency.

Spooling functions that provide enhanced input and output operations for batch jobs. All input jobs from diskette, card, or data base are 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 output spooling function, in which the output of a batch job to the printer, diskette, or card punch is written to auxiliary storage, does not limit job execution to the availability or speed of these devices. This capability 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 printing on a work station printer the output can be redirected to the system printer.

Security functions, which are increasingly important in a work station environment, allow protection of data and other system resources so that unauthorized persons do not have access to them.

The user can specify the level of security level. This can range from a minimal level in which all defaults are taken and files are simply protected from inadvertent destruction to a very secure system.

The security function also allows an individual (such as the DP manager) to define for each user of the system a personal *user profile* that uniquely identifies that user to the system, requires the use of his own password, and allows him to be specifically authorized to certain files and programs.

Application development aids such as the ability to store text information in the data base to describe each file, record, and field. This information provides a current level of detailed documentation to the user that he can display or print whenever it is needed. Also, a file reference function provides extensive cross reference information about the user's programs, files, record formats, and fields.

Service functions are improved over previous systems by both an increase in the number of available aids and an improvement in the user interface. Since these aids and diagnostic programs run under control of the Control Program Facility, the service representative can generally check input and output devices or diagnose problems on the system concurrent with user operation.

Message handling functions that provide extensive support for:

- Error notification
- Logging activity
- Sending messages (such as, to the system operator, between work station users, and to user message queues)

System error messages consist of first and second level messages. The first level presents the user with an error message number and a brief description of the error condition. A user needing further information can request the second level message which further describes the error and gives corrective information. Therefore, the need for the user to look up error information in message manuals is substantially reduced.

The message handling function provides extensive new features for developing applications in the work station environment 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 which 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.

RPG III

RPG III includes many new language functions, such as:

- Externally described data that provides ease of coding, consistent naming conventions, and improved application growth capability.

- Full procedural file specification that allows the user to process the same file in both a random and sequential manner in the same program. This file specification allows many common applications to be developed with greater ease.

- Program structures that do not use the RPG III cycle but allow read, write, update operations to the data base and special work station input and output operations to be specified directly in the calculation logic. This program structure provides a more straightforward approach to interactive applications and an alternative for complex batch programs.

- The capability to allow any separately compiled RPG III program to be called by another program and also call other programs along with the passing of parameters. These programs can be written either in RPG III or control language, thereby enabling a programmer to easily construct a complete application consisting of RPG III and control language programs. There is no requirement to combine these programs in any manner (such as a link-edit operation) prior to execution.

- New logic control operations, such as various forms of the DO and the IF operations.

- Data structures to allow related data to be manipulated either as individual data fields or as a group.

Interactive Data Base Utilities (IDU)

The System/38 Interactive Data Base Utilities are a set of utilities that support interactive definition and execution of the following functions:

- Creating and maintaining program language and data description source files.
- Entering, displaying, and maintaining data files.
- Extracting, summarizing, and presenting information from data files.

The Source Entry Utility (SEU) is used to enter and maintain source data. For example, source statements for programs (RPG III and control language) and data description specifications can be entered from a work station through the Source Entry Utility. With the Source Entry Utility, the programmer is helped by the use of prompting, format control, automatic statement sequencing, syntax checking, and the ability to view several statements of a program while working with an individual statement.

The Data File Utility (DFU) allows the user to quickly define data entry, inquiry, and file maintenance applications without the need to write a program. On previous systems, if a user wanted to define a data file utility application, he needed 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. However, with the System/38 Data File Utility, this application definition is greatly simplified by the fact that the data fields to be used by that application have been previously described in the data base. Therefore, the user need only name the file to be used.

The Query Utility allows the user to interactively specify criteria for the extraction, summarization, and presentation of information from the data base. For example, a table presenting dollar sales and percent of total sales by product group and sales territory can be generated. Consequently, many applications that would have required considerable effort to program using RPG III can now be quickly and easily specified with the use of the Query Utility.

System/3 Batch Conversion Utilities

The System/3 Batch Conversion Utilities for System/38 are available to assist the System/3 user in converting his RPG II programs, operational control language, and data files to System/38. While conversion can be achieved without these utilities, their use significantly shortens the time required to convert from one system to another. The System/3 Batch Conversion Utilities are designed to allow the user to do as much of the conversion process as possible on his System/3. Therefore, the user can accomplish the actual move to System/38 very quickly, and minimize the disruption of his data processing operation.

Conversion Reformat Utility

The System/38 Conversion Reformat Utility creates a tag file or a reformatted physical file. The Conversion Reformat Utility duplicates all of the functions of the System/3 Sort, except summary sort. 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 is provided to more closely duplicate the System/3 Sort Utility and enable a quicker conversion from System/3.

MACHINE

System Unit

The System/38 system unit contains a processing unit, a main storage capacity, a system console with keyboard and display screen, a diskette magazine drive, a fixed-disk auxiliary storage capacity, plus a work station controller. All of these components are contained in a small attractive enclosure that has sharply reduced energy requirements for power and cooling.

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 communications between main storage and input/output devices.

The IBM 5381 Processing Unit offers up to 1536K bytes of main storage. The fixed disk auxiliary storage offered ranges from a standard 64.5 megabytes to up to 387 magabytes with five additional disk enclosures. Both main storage and auxiliary storage utilize IBM advances in the technology of nonremovable, sealed, direct access, auxiliary storage assemblies. Advances in this type of storage make possible improved price/performance levels, and improved efficiency and security because there are no disk packs or data modules to mount or remove from the system. Reliability is also improved because there are fewer mechanical parts and less restrictive tolerances.

The machine provides storage management based on a uniform level of addressing for all data, programs, and files. This storage management treats all auxiliary storage as simply an extension of main storage. Therefore, the Control Program Facility as well as user programs can exceed the amount of available main storage. This type of virtual storage allows the Control Program Facility to 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 is designed to relieve the programmer of the need to plan his applications so that related programs are in main storage at the same time.

Storage management does not partition main storage, nor is there any requirement for the programmer to code his programs within specific sizes to provide better performance or to fit everything into certain size task areas. All main storage is shared with the exception that the batch and the interactive jobs will normally be assigned separate pools of main storage to prevent the batch jobs from utilizing main storage so heavily that the interactive jobs do not obtain a desired response time. The user has the facilities 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 from any available location on all the attached auxiliary storage.

The system console is 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 conditions in the process on the system, switches to control system operations, and keys to power the system on and initiate the microprogram load process.

The diskette magazine drive provides the capability to 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 of removable data.

Input/Output Devices

System printing is performed with the IBM 5211 and the IBM 3262 printers. They are floor-standing units that are quiet in operation and easy to set up. Both use a flexible steel print belt that can be quickly changed by the system operator to accommodate changing printing requirements. One model of the 5211 printer is available with a speed up to 300 lines per minute. Two models of the 3262 printer are available with speeds of 650 lines per minute.

Many standard and special purpose print belts are available, including a 60-character print belt which optimized to a statistical analysis of commercial printing requirements. This print belt provides better performance than a non-optimized print belt and supplies all the special characters required for most users, thereby reducing the requirement for print belt changes.

The IBM 5424 Multi-Function Card Unit 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 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 with the faster model having the capability of reading 500 cards per minute and punching and printing 120 cards per minute.

The IBM 3410/3411 Magnetic Tape Subsystem is available for System/38 users who are growing from systems equipped with a 9-track magnetic tape input/output device. The 3410/3411 subsystem for System/38 consists of one 3411 Magnetic Tape Unit and Control. One to three additional 3410 Magnetic Tape units can also be attached.

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

The display stations and printers can be set up 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, maintenance can be carried out without disrupting other areas of the system.

The display stations available with this system offer two sizes of display screens, 960 characters or 1920 characters, a typewriter-like movable keyboard with numeric keys, and a command shift key for selecting a number of command functions.

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

The System/38 work station controller allows direct attachment of up to 40 work stations within the same local environment as the system. There is no need for common-carrier-provided services to attach the 5250 devices. System/38 can also communicate with work stations in remote locations through a variety of attachment methods.

Chapter 2. System/38 Program Products

IBM has four program products available to help the user efficiently operate System/38 and develop applications. They are:

- Control Program Facility
- RPG III (an application programming language)
- Interactive Data Base Utilities
- Conversion Reformat Utility

These program products will execute on all models of System/38. Therefore, all system users have the same *functional* capability whether they have the smallest or the largest system.

CONTROL PROGRAM FACILITY (CPF)

The System/38 Control Program Facility is the major user interface with system functions. It has been designed to provide a much greater range of function than prior control programs. While all users may not immediately require the full range of function provided by the Control Program Facility, all of the function is always available whenever the user needs it. The user may select whatever function he needs at any time. No system specialization or selection of Control Program Facility functions is required, even at the installation of the Control Program Facility.

For ease of use, the most common options are assumed by default and need not be specified by the user unless there is a special or unique requirement. For example, it is assumed that a printed report should be produced on the system printer on standard paper unless the user specifies that the report should be printed on a work station printer using a special preprinted form that requires alignment.

The most commonly defined objects (those which are necessary for an executable system) are provided by IBM. These include:

- Work station message queues.
- Subsystem definitions.
- Job queues and output queues.
- A general purpose library.
- Source files for the most common types of source data.
- A default security definition that includes standard user profiles for the work station user, programmer, system operator, and security officer.
- System values for such things as the history log size and data format.

The use of defaults and supplied definitions for system functions allows the user to install System/38 quickly and with a minimal amount of initial training. Then as the user becomes more familiar with the many options available, he can select the specific options he wants to improve his operation.

The major areas of function provided by the Control Program Facility are shown in Figure 2-1. What is shown represents areas of functional support as the user tends to view them, rather than the internal components of the Control Program Facility.

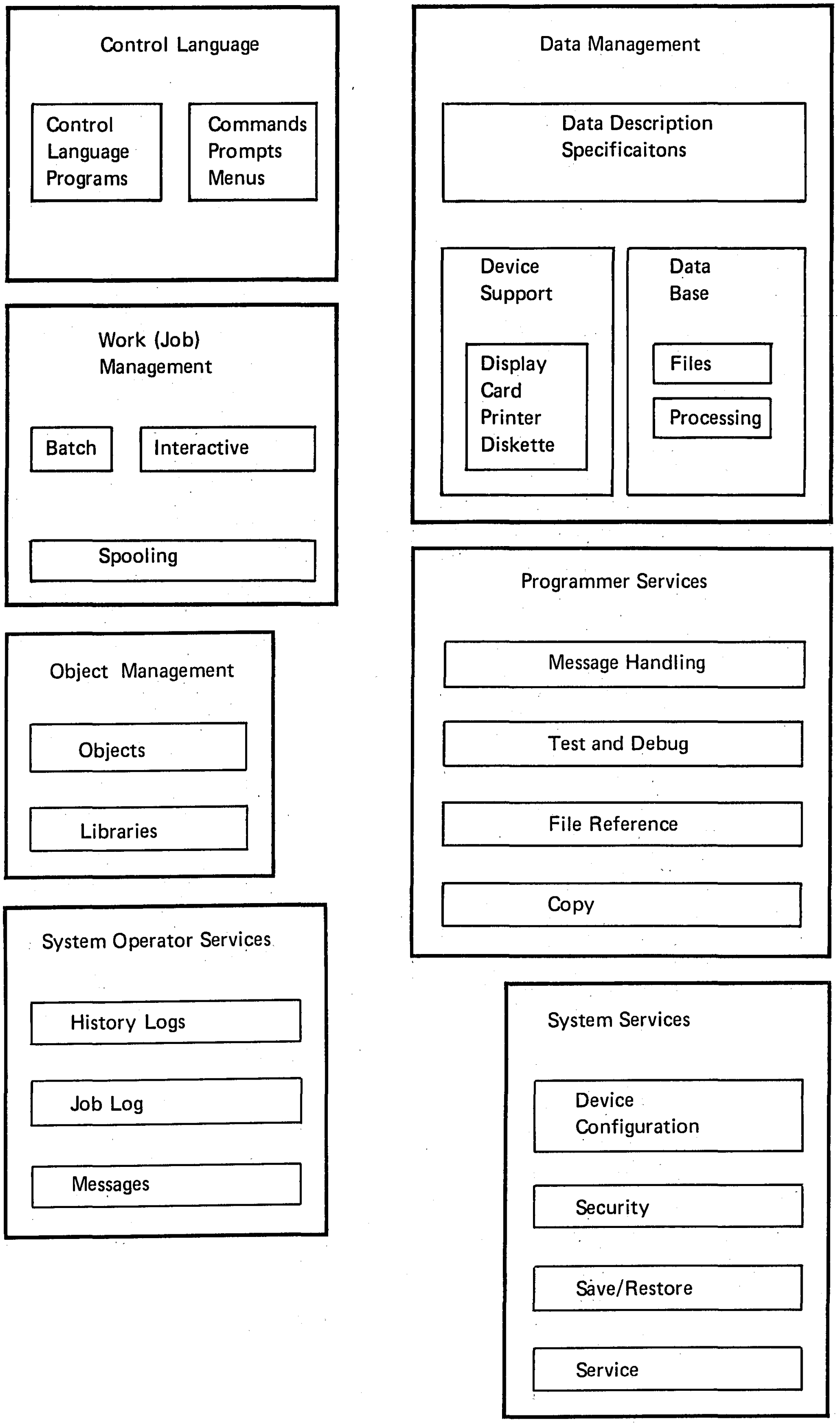


Figure 2-1. Control Program Facility Functions

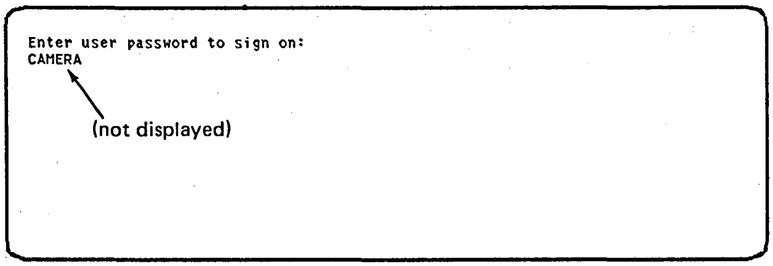
User Interfaces

The Control Program Facility provides the user with three interfaces to the system functions:

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

These interfaces are normally used by the system operator and the programmer 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. Generally, the work station user is not expected to learn the control language and the data description specifications.

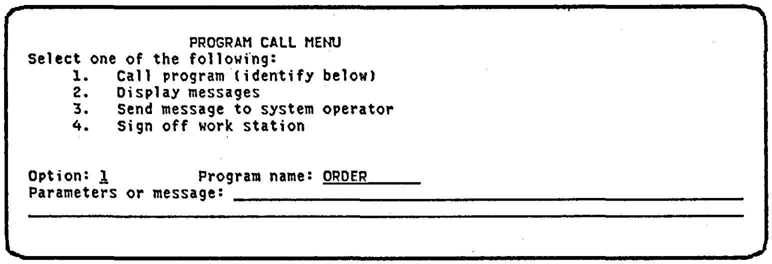
A typical series of displays that the work station user (such as an order entry clerk) might see are:



```
Enter user password to sign on:
CAMERA
```

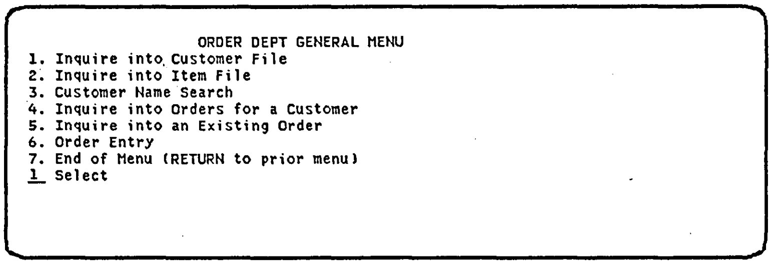
(not displayed)

Control Program Facility 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.



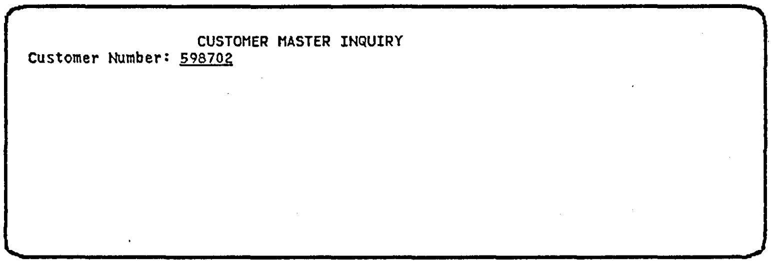
```
                    PROGRAM CALL MENU
Select one of the following:
     1.   Call program (identify below)
     2.   Display messages
     3.   Send message to system operator
     4.   Sign off work station

Option: 1           Program name: ORDER_____
Parameters or message: ________________________________________
_______________________________________________________________
```

Control Program Facility 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.



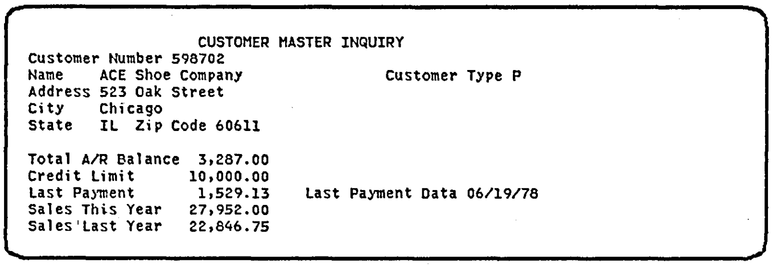
```
                        ORDER DEPT GENERAL MENU
1. Inquire into Customer File
2. Inquire into Item File
3. Customer Name Search
4. Inquire into Orders for a Customer
5. Inquire into an Existing Order
6. Order Entry
7. End of Menu (RETURN to prior menu)
1  Select
```

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
```

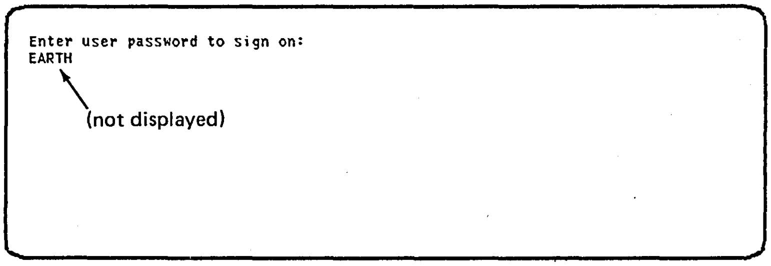
User Created Display
A prompt requesting input of a customer number is presented. The user responds with 598702.



```
                        CUSTOMER MASTER INQUIRY
Customer Number 598702
Name    ACE Shoe Company                Customer Type P
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 Data 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.

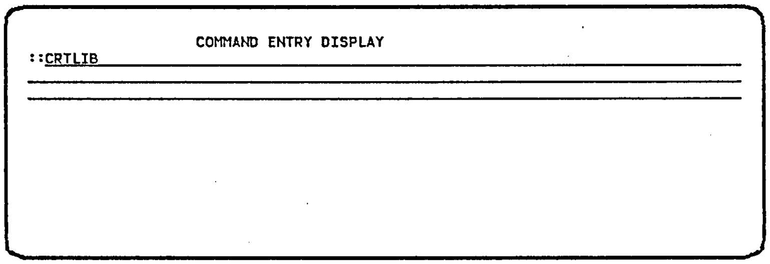
A typical series of displays that the programmer might see are:



```
Enter user password to sign on:
EARTH
```

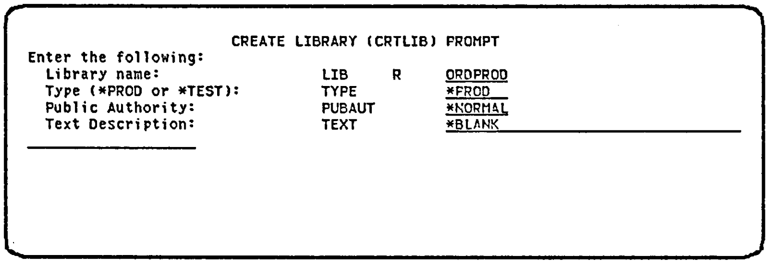
(not displayed)

Control Program Facility 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 Command Entry Display whenever he signs on.



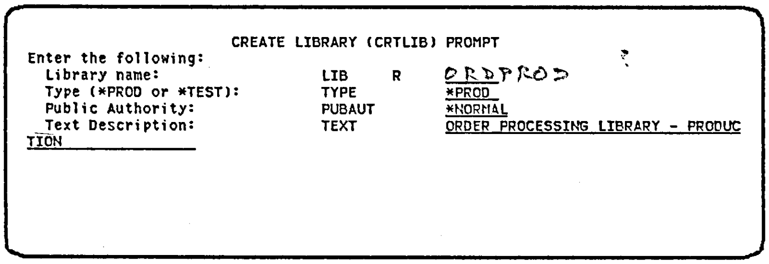
```
                    COMMAND ENTRY DISPLAY
::CRTLIB
```

Control Program Facility Supplied Display
The Command Entry Display allows the user to enter control language commands. In this example, the user 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). Although he knows the command he can not remember the parameters that go with the command. Therefore, he requests *prompting* by pressing command function key 3.



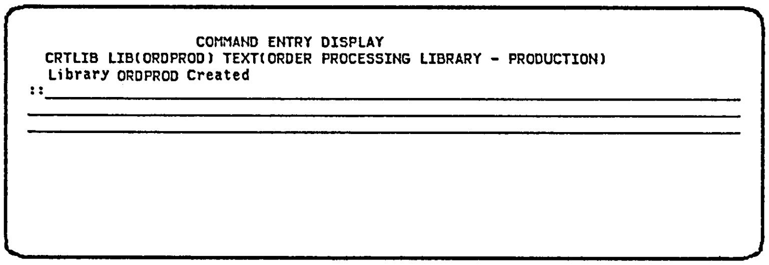
```
                        CREATE LIBRARY (CRTLIB) PROMPT
Enter the following:
  Library name:                  LIB     R     ORDPROD
  Type (*PROD or *TEST):         TYPE          *PROD
  Public Authority:              PUBAUT        *NORMAL
  Text Description:              TEXT          *BLANK
```

Control Program Facility Supplied Display
The user 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
  Type (*PROD or *TEST):         TYPE          *PROD
  Public Authority:              PUBAUT        *NORMAL
  Text Description:              TEXT          ORDER PROCESSING LIBRARY - PRODUC
TION
```

The user 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 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.



```
                    COMMAND ENTRY DISPLAY
  CRTLIB LIB(ORDPROD) TEXT(ORDER PROCESSING LIBRARY - PRODUCTION)
  Library ORDPROD Created
::
```

Control Program Facility Supplied Display
The command is executed. The user is returned to the Command Entry Display where he sees a display of the command he just executed. He also sees the system response, ORDPROD Library Created. He is then free to enter another command.

Control Language

The new 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 in a variety of ways. They 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.

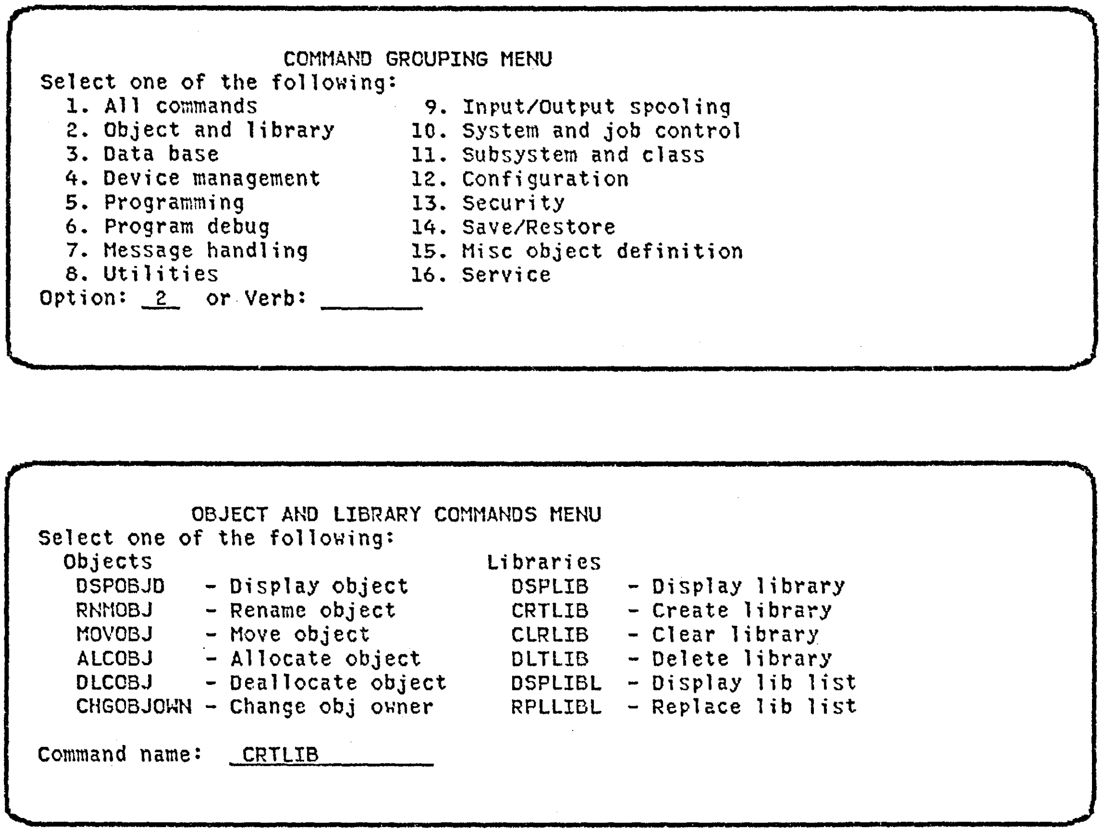
For example, to send a break message to all the work stations, the following command is used:

$$\underbrace{\text{SNDBRKMSG}}_{\substack{\text{Command}\\ \text{Name}}}\ \underbrace{\text{MSG('Please Sign Off')}}_{\text{Parameter}}$$

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 the user in entering the parameter values. (The two Create Library (CRTLIB) prompts are examples of prompts a programmer might see.) This feature can be especially helpful when part of a command is known but help is needed to remember certain parameters. When a prompt is requested after part of a command has been entered, any parameter values that have already been entered are shown on the prompt. Every command is available for direct execution from a prompt. When a command is entered, it is automatically syntax-checked. For example, if an error is made entering a command, an error message is displayed. The user needing more information about the error can request a display of a second-level message by pressing the Help key.

If the user is not sure which command to enter, the Control Program Facility provides a set of menus that help the user 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. A selection of one of the command groups, such as 2, results in the display of a list of commands for the selected area (for example, the object and library commands menu). A command name is entered on this display and the prompt is presented just as if the user 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              9. Input/Output spooling
  2. Object and library       10. System and job control
  3. Data base                11. Subsystem and class
  4. Device management        12. Configuration
  5. Programming              13. Security
  6. Program debug            14. Save/Restore
  7. Message handling         15. Misc object definition
  8. Utilities                16. Service
Option: _2_  or Verb: ________
```

```
            OBJECT AND LIBRARY COMMANDS MENU
Select one of the following:
  Objects                           Libraries
   DSPOBJD   - Display object        DSPLIB   - Display library
   RNMOBJ    - Rename object         CRTLIB   - Create library
   MOVOBJ    - Move object           CLRLIB   - Clear library
   ALCOBJ    - Allocate object       DLTLIB   - Delete library
   DLCOBJ    - Deallocate object     DSPLIBL  - Display lib list
   CHGOBJOWN - Change obj owner      RPLLIBL  - Replace lib list

Command name:  _CRTLIB_______
```

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. Control language commands can be used to write a program, and that program can then be compiled and assigned a name by which it can be executed by the user or by another program. Compiling provides much better execution performance and allows more flexibility in the use of the language, such as 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.

The use of control language programs is more powerful than the grouping of control statements into procedures. An example of a small control language program is:

```
/* 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)
  THEN CALL CRDAPRV PARM(&ORDNO) /* CALL CREDIT APPROVAL PGM */
  ELSE CALL ORDPROC PARM(&ORDNO) /* CREDIT IS OK. PROCEED */
ENDPGM
```

Control language programs can call or be called by RPG III 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 programs and RPG III programs.

Batch Job Streams

Control language commands are also used in batch job streams that can be entered from the diskette magazine drive, a card reader, 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
    •
    •
    •
// ENDDTA
```

Object Management

Objects

The term *object* does not just refer to a compiled program. Object is a general term used to refer to many different things on System/38, such as a program, a file, a message queue, or a user profile.

Object generally refers to related data that has a name and can be operated on with control language commands (for example, to create, change, delete, or display). Briefly, each object has a set of attributes that are assigned to it. These attributes are:

- Name—I ne 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 are contained in libraries. Any library can contain 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. This can be done without making any changes to the application or the control language programs that name the objects used in the application. Only the name of the library to be accessed by the application is changed.

An example of this use of libraries is in testing an application. A set of data files with test data and programs can be used by the application 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. 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. If a library is secured, all the objects contained in that library are protected.

Any number of libraries can exist on the system. As Figure 2-3 illustrates, some libraries are provided by the system, others can be created by the user. 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.

- A *general purpose library* that is a default library where user created programs and data are placed unless specifically placed in another library. Any authorized user can use this 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. Therefore, multiple work station users each have their own temporary library even if they are executing the same application.

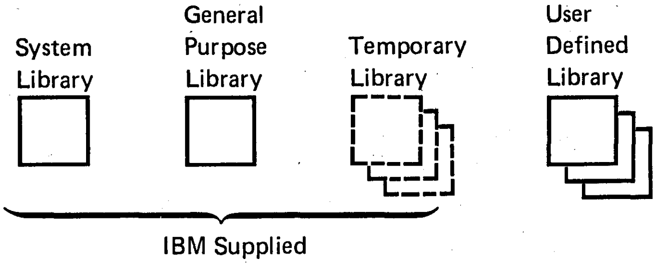


Figure 2-3. Libraries

In addition to maintaining the libraries that the system provides, the programmer can define, rearrange, and delete any number of user-defined libraries. These libraries are either production or test libraries.

Managing Work

One of the main measures of performance in batch processing is throughput, the amount of work handled in a certain period of time. The major factors that influence system throughput are:

- Processing speed and its use.
- Main storage and its use.

Both these factors are managed by automatically controlling the flow of jobs through the system. The Control Program Facility accepts requests from the system users, obtains the resources needed to satisfy the requests, and schedules the work to be done according to the priority assigned to the requests.

Some other factors are:

- Spooled versus nonspooled input and output.
- Input/output speeds.
- Job mix.

Although the Control Program Facility manages the work flow, the user is responsible for indicating to the system how that work should be managed. Through the control language, the user can establish priorities for work done on the system. Once these priorities are established, the system uses them as it manages the work flow on the system.

Jobs

There are two types of jobs on System/38, batch and interactive.

A batch job can generally be thought of as being delimited by a //JOB command and a // ENDJOB command. The batch job is normally placed in a job queue to await execution, and during execution the job has no interaction with a work station user.

Batch jobs can be initiated by the work station user or a control language program by using the Submit Job command which causes a batch job to be placed in a job queue.

An interactive job is simply all of the functions performed between sign-on and sign-off at a work station.

Subsystems

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

The subsystems provided with the Control Program Facility are designed to supply generalized environments that meet the characteristics and requirements for the processing of interactive jobs, batch jobs, and input and output spooling operations. The subsystems provided are:

- Two copies of the *interactive* subsystem that support interactive processing through all the work stations attached to the system.
- A *batch* subsystem that supports batch job processing on the system.
- A *spooling* subsystem that supports jobs that read batch job streams into the system from input devices and jobs that write output files to output devices.

However, if these generalized environments do not meet a particular user's requirements, control language commands can be used to tailor a subsystem to the user's needs or to create other subsystems.

The interactive subsystems support work stations. One of these subsystems is activated when the system is started and remains active until the system is terminated. All work station devices, including the system console, are attached to this subsystem. Two copies of the interactive subsystem are provided to allow one copy to be inactive (enabling it to be changed) while the system is running.

The batch subsystem supplies batch work support and must be active only when batch jobs are to be processed. 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.

The spooling subsystem schedules and controls the system input and output through unit record devices (diskette, card, printer). Input can also be spooled from the data base.

In addition to the supplied subsystems, a user can define his own subsystems. In this way, programmers can design work environments to execute specific types of applications.

For any subsystem, work is selected for execution based on information supplied by the user when he defined the subsystem. Selection is also based on information obtained from the system, relative to its ability to handle the current work load, and on the special instructions supplied by the system operator.

Spooling

Two forms of spooling exist on System/38, 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 it in an input queue.

- Output spooling is the function associated with output queues. Output spooling allows programs to write to an output queue rather than directly allocating the output device. Output spooling also offers flexibility in terms of which device can be used and when the output operation actually occurs.

Spooling has the ability to 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 printing of an output file.

With spooling, the user can generally decrease the length of time required to execute a job and increase the number of jobs that can be executed in series while optimizing the throughput of input and output devices.

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

- Performance is significantly improved over nonspooling environments under most circumstances. Because of the efficient use of input and output devices, job execution is not bound by the availability or speed of the device. This permits the devices to be run at their rated speed for more optimum use. Therefore, 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 the programmer does not have to concern himself with device contention (for example, when two or more concurrently executing programs may require the use of the same device, such as a printer or card input/output device).

- System design is also much easier for those jobs where input from multiple work stations is destined for a single output device. The spooling function can treat the output created from each work station as a separate file.

- Since the program is not waiting for slow device completions, it will complete faster with better use of the system resources.

- With local and remote device spooling support, the same set of application programs may accept input from or direct output to the local devices and the remote work stations. On System/38, the entire spooling operation can be controlled by the work station user with little or no involvement by the system operator.

- Operation considerations are more easily handled with spooling. For instance, rush jobs can be brought to the head of the input or output queue.

Data Description Specifications (DDS)

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

On System/38, data description specifications allow the user to describe the data fields external to the program (Figure 2-4). The description of each field (name, data type, length, decimal positions) and the order of that field within the record are defined through the data descriptions specifications and stored with the file in auxiliary storage.

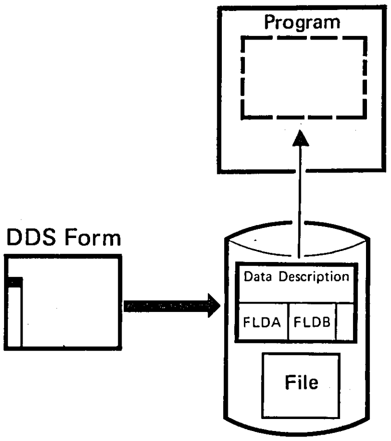
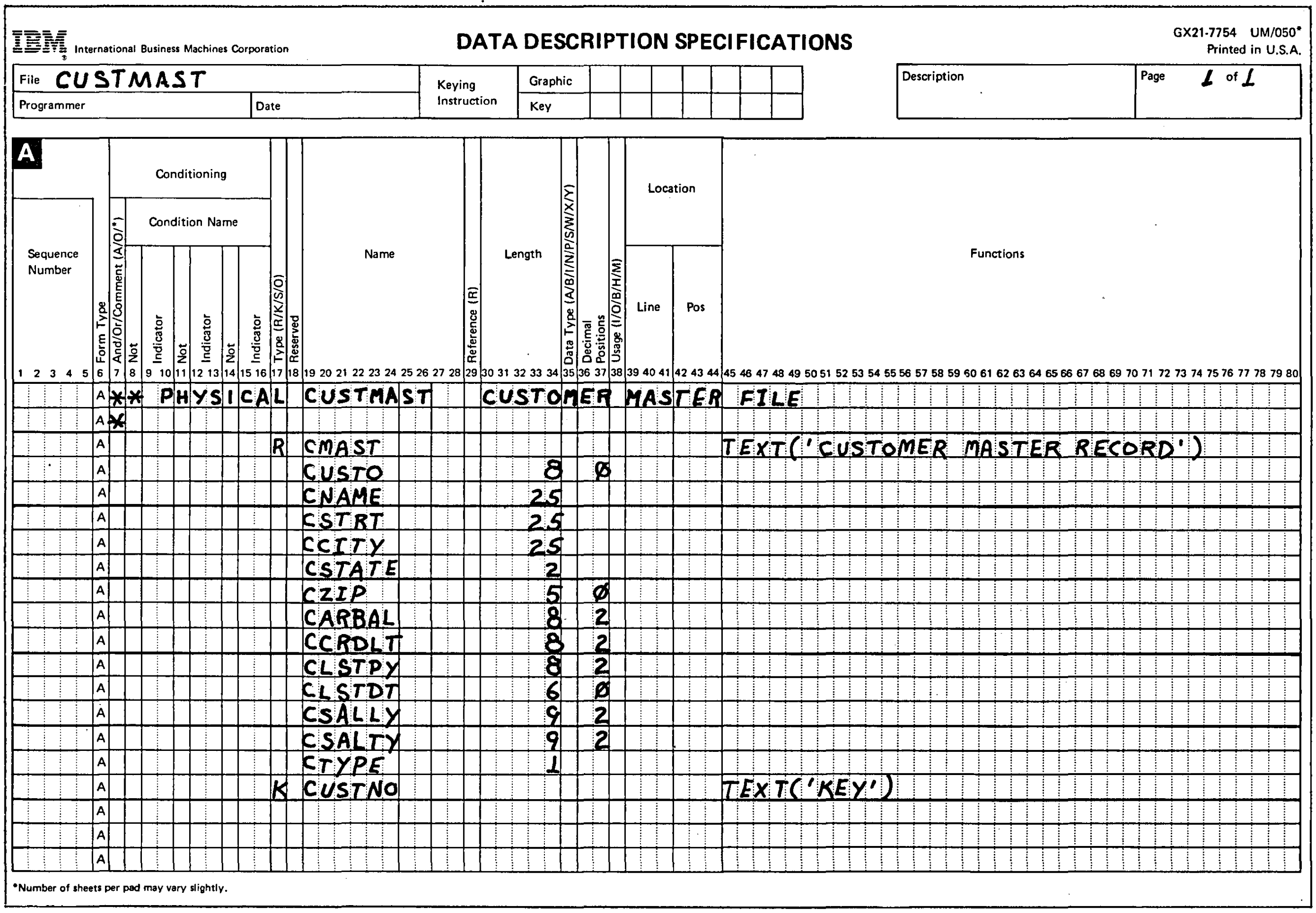


Figure 2-4. Externally Described Data

Data description specifications are also used on System/38 to describe record definitions for devices such as the work station. In addition to the field attributes of name, data type, and length, the work station data descriptions include many other important attributes, such as:

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

Data records are described by using the data description specifications form (Figure 2-5). The data description specifications are normally entered interactively from a work station using the Source Entry Utility which provides prompting, format control, and syntax checking.



IBM International Business Machines Corporation

DATA DESCRIPTION SPECIFICATIONS

GX21-7754 UM/050*
Printed in U.S.A.

File CUSTMAST | Keying Instruction | Graphic | Key | Description | Page 1 of 1
Programmer | Date

A

Sequence Number | Form Type | Conditioning: And/Or/Comment (A/O/*), Condition Name: Not, Indicator, Not, Indicator, Not, Indicator | Type (R/K/S/O) | Reserved | Name | Reference (R) | Length | Data Type (A/B/I/N/P/S/W/X/Y) | Decimal Positions | Usage (I/O/B/H/M) | Location: Line, Pos | Functions

```
1 2 3 4 5 6 7 8 9 10 ... 80
     A** PHYSICAL CUSTMAST   CUSTOMER MASTER FILE
     A*
     A          R CMAST                     TEXT('CUSTOMER MASTER RECORD')
     A            CUSTO          8  0
     A            CNAME         25
     A            CSTRT         25
     A            CCITY         25
     A            CSTATE         2
     A            CZIP           5  0
     A            CARBAL         8  2
     A            CCRDLT         8  2
     A            CLSTPY         8  2
     A            CLSTDT         6  0
     A            CSALLY         9  2
     A            CSALTY         9  2
     A            CTYPE          1
     A          K CUSTNO                    TEXT('KEY')
     A
     A
     A
```

*Number of sheets per pad may vary slightly.

Figure 2-5. Data Description Specifications Form

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 specifications. The data description specifications are used to describe display formats, as well as data base formats.

Several benefits result from separating data descriptions and programs.

- There is a reduction in the amount of coding required in the high-level language (RPG III) programs since the input and output specifications (data description) for data base files and display formats are not coded in the program. (The data description is required only once, in the file.)

- With the use of externally described data, a good level of data independence can be achieved in which changes to the data attributes, or the addition of new data fields to a file, often do not require changes to the programs using these files. Frequently, the need to recompile many of the related programs is eliminated.

- 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 may 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.

- 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

  Tnis 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 layouts are no longer an important design consideration. Each field in the record is described, including the length of the field. The order in which the fields are described determines their order in the record. The specific character position of the field in the record is not specified nor is the record length. Different record types in the same 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 as well as the description of the file is always stored.

The ability to describe files and record formats (using the data description specifications) outside the application program lets the programmer 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 users can simultaneously access, update, add to, and delete records in the same file. To provide integrity, an individual record is locked (unavailable to other jobs for update) when a user's job retrieves it for update.

- Described data–All data is known to the system. Definition of unique physical files can occur down to a field level. Each field can be completely described with 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–Most applications require access to a set of data (a data base file) by more than one key or index. For example, the user may 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, the user designs a secondary index and writes data management code in the application to access the data. System/3 supports multiple indexes to the data. The indexes are called access paths on System/38 because the method in which they are implemented is quite different than on other systems.

  The multiple access paths that can be defined for a file can all 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 2-6).

  Selection criteria can be applied when defining an access path. For example, a programmer may wish 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 the programmer creates a data base file, he specifies 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.

Data can be accessed 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 the user's 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 maintain complex user-created indexes or chains of relative record numbers.

The user's data can be accessed through both the logical file and the original physical files. The programmer can define as many logical file relationships as he wants, any time he wants. The logical files look just like physical files to the programs; no special language is required to access them.

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 previous systems. A physical file contains fixed-length records, all of which have the same format.

With physical file descriptions, the programmer 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 or keyed sequence.

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.

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 physical representation of the data base.

Any number of logical files can access the same physical file (Figure 2-6). 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 rather 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.

This capability, along with the ability to define multiple access paths, 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 is a significant aid in reducing the number of compilations required when data base changes occur.

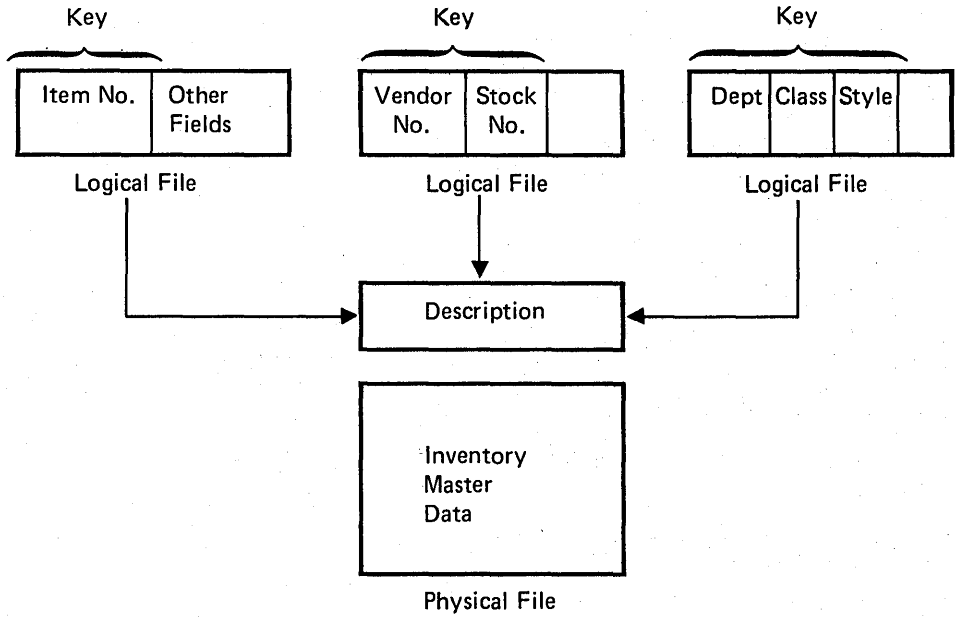


Figure 2-6. Multiple Logical Files Using a Single Physical File

Another benefit of logical files is that a logical file can be defined over two or more physical files (Figure 2-7) resulting in a logical file with multiple record types. There is no need for these different record types to be the same length (generally, they would not be). This results in a much simpler file design than on prior systems in which each of the different record types in the file had to be the same length, usually forcing the programmer to assemble and disassemble (block and deblock) records in the programs.

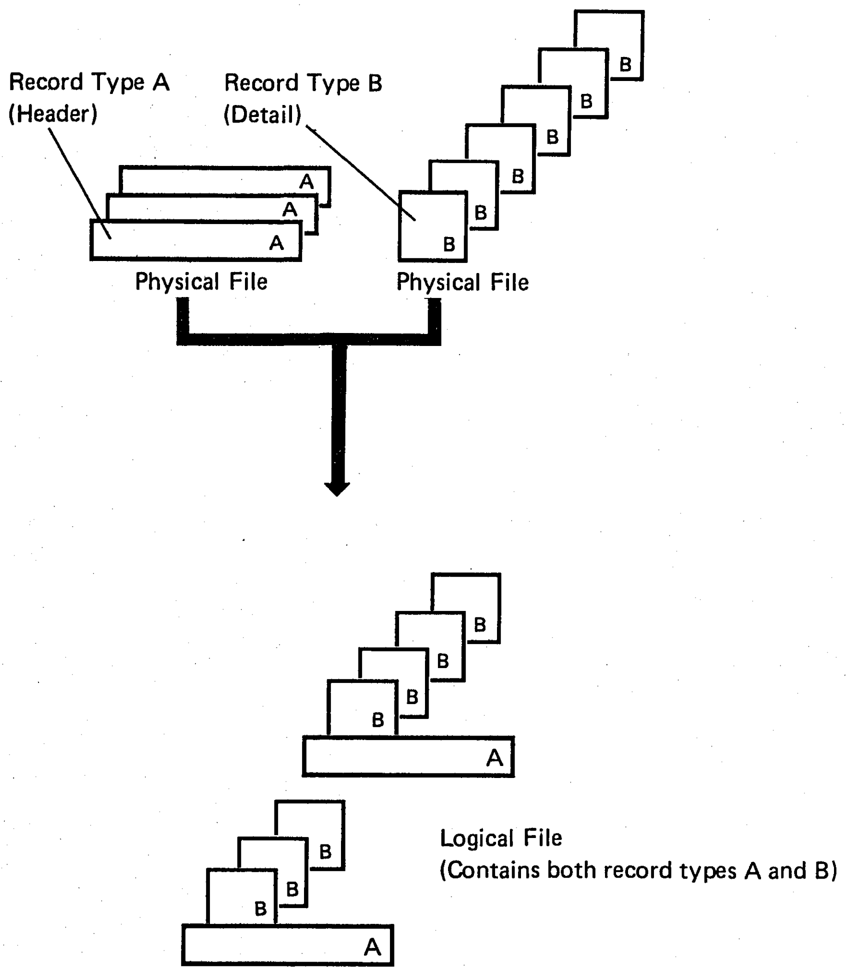


Figure 2-7. A Single Logical File From Multiple Physical Files

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

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

Many other file operations that simplify application development are provided by the System/38 data base data management.

Device Data Management

Device data management uses device file descriptions (work station, printer, card reader) stored in the system but the data comes from, or is sent to, a device when the file is processed.

Device data management consists of two major areas:

- Display (work station and system console) device data management which generally uses externally described data.
- Nondisplay device (diskette, card, printer) data management which generally uses program-described data. The printer can optionally use externally described data.

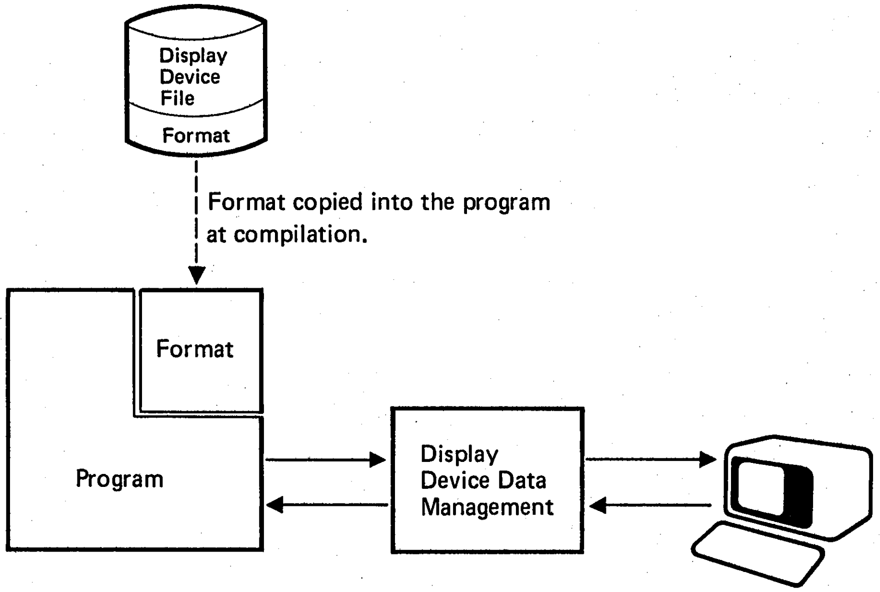
Display Device Data Management

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

- Position of the field on the display.
- Type of field (input, output, or both).
- 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 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 keyboard locking and unlocking.

All of these functions are provided by display device data management, thereby greatly reducing the amount of coding required to write work station applications. Most of this function is specified in external data description specifications which are stored in a display device file (Figure 2-8). For example, an input field may have validity checking which specifies the field is mandatory entry, with a value between 1 and 5. If no data is entered, or anything other than a number from 1 to 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 of this is accomplished without any user application programming.

The user's program communicates with the display device file with the use of conditioning and response indicators in the data description specifications that can be set and read in the program. For example, setting on conditioning indicator 02 in the program may cause several optional fields to be included with the output to the display. Also, the entry of data in a specified field may cause response indicator 05 to be set on when the program reads from the display.



Display device data management performs the output and input operations requested by the program. Information in the format is used to control the input and output operation.

Figure 2-8. Display Device Data Management

The Control Program Facility performs all the operations needed to control the work station, including formatting data on the display screen, handling error conditions that occur at the station, and performing the input and output operations at the station. In addition, this system support lets the programmer create application programs that are independent of which work station is involved in the input and output operations.

Also, System/38 provides a subfile function. A portion of the display can be described like a page of similar records from a file (for example, a page of line items in a customer order, the transactions against a checking account statement, or the charges on a customer's bill). The work station user treats that portion of the display as a long list of records that can be rolled up and down. Changes can be made to the data as it is displayed. The application program then reads the entire subfile, processing one record at a time, reading only those records which have been changed. The subfile function permits a major reduction in the effort to develop many common applications.

Device file descriptions let the programmer produce display screen definitions that can be used directly by RPG III programs. Therefore, the programmer defines the display screen once and does not need to redefine that display in the input or output specifications of the application program.

Diskette, Card, and Printer Device Data Management

With diskette, card, or printer files, the device file generally describes only the file characteristics. The data fields are described in the user's program. The system allows a considerable amount of flexibility in switching the input or output of a program from one device type to another. For example, output written for the diskette can be printed. All of the special file characteristics can be specified in the device file. For 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 of the functions required for online program development from either local or remote work stations.

- Online source entry (using the Source Entry Utility).
- Ability to submit a job for compilation from the work station.
- Ability to display the output of a compilation at the work station, or print it on the system printer or on the work station printer.
- Execution time test and debug facilities.
- Ability to move data between device files or data base files.

Program development can also be accomplished using the traditional batch methods.

Testing and Debugging

Many System/38 functions are useful in testing, such as 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 functions specifically for testing and debugging:

- Test library/test environment

- Debug function

When testing an application, the user can operate in a test environment. When in the test environment, the user's programs can only update files (objects) in a test library. Therefore, all objects in production libraries are protected from alteration. By use of the library list in which the name of a test library is placed ahead of a production library, a test can be performed in which test programs and files are used from a test library. Other programs and files are used from a production library. Testing can be performed concurrently with normal production with no risk of destroying production data.

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

The debug function, which is normally used in conjunction with the test environment, allows the programmer to specify breakpoints in control language programs or RPG III programs. Breakpoints are points in a program at which execution is suspended, allowing the programmer to display and alter variables and indicators. Breakpoints, variables, and indicators are specified in terms of control language or RPG III source (for example, the statement number and the variable name).

The debug function is used at execution time and allows the programmer to examine the operation of a program in complete detail. Nothing is specified at the time the program is compiled. While a program is suspended at a breakpoint, the programmer can use the System Request key to perform some other function. For example, he could stop a program right after the point where it updates a file. Then he could use the Data File Utility to display a record in the file to see if the update was performed properly. Then he could resume execution.

Interactive debugging would normally be used to debug both batch and work station programs. Also, a trace function provides a listing of the Interactive debugging would normally be 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. The programmer can use this function to provide several types of cross reference information. The file reference function provides information about the:

- Definition of the data base.
- Organization of the data base.
- Usage of the data base.
- Definition of device files.
- Usage of device files.

The information is extracted 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 is highly useful for the following:

- Designing new applications which must interface with existing applications and files.
- Making changes to existing applications.
- Finding errors in existing applications.
- Providing supporting documentation for 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 are dependent upon a specific physical file.
- A list of all files referenced in a program (or group of programs).

Through optional parameters the user can build a data base file containing detailed information about each data field in a file or group of files. This information includes all the attributes of the field and any references made by the field (for example, other fields upon which the attributes of this field are based). Normally, this data base file would contain a very large amount of detailed information. Therefore, the programmer would use a logical file over this file or use the Query Utility to select and display information from this file as it is required.

The file reference information can be used to answer questions, such 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

All messages generated by the Control Program Facility and other IBM program products are stored as data in separate message files and can be manipulated externally to the function that requests them. Therefore, 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 system-provided message queues. The message queue then delivers the message to the work station or job associated with the queue. The message queue delivery operation can be altered by the programmer to obtain the following effects:

- Immediate delivery.
- Notification of the arrival of a message.
- Holding of the message for later delivery.
- Rejection of the message.
- Use the default reply.

A permanent message queue is provided for each work station, the system console, the system operator, and the history log file. In addition, the system allows the programmer to:

- Create message queues and message files.
- Create, store, and retrieve messages.
- Route messages to various locations in the system.
- Queue the delivery of messages at various locations.

Copy

The copy function can be used to move data between combinations of device files or data base files. Unlike previous copy utilities, the System/38 copy function is field-level sensitive. Therefore, if the copy function moves data between two data base files, the fields can be rearranged, enlarged, or dropped. A method of defining fields for device files also exists.

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

System Operator Services

The system console is the primary interface device for the system operator. While System/38 is designed as a work station system that allows work station users to perform work without the need for a system operator, some functions still will be periodically performed by the 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.

Since control language can be entered at any work station (local or remote), many of the system operator functions can be performed at any work station. In addition, 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. An option also exists that permits the system to be started even if the sytem console is inoperable. In this case, one of the work stations can be used by the system operator.

Display commands are provided to display objects, entries in libraries, and status information. For example, the spooling function gives the system operator the capability to 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 system operator commands executed and all messages sent to the system operator. The history log supplies an audit trail of the activities performed on the system and can be printed or displayed on command.

Job Log

The job log supplies the user with 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.

A job log is created by the Control Program Facility at the start of each job. For batch jobs, creation of the log occurs when the job is read in. For interactive jobs, creation occurs during sign-on at the work station. The job log is printed with the job output for batch jobs and printed optionally for interactive jobs.

Messages

The message handling function of the Control Program Facility allows either automatic responses or 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 to be 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 to manage the operation of the system. These functions include:

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

Device Configuration

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

The primary requirement for the installation of the Control Program Facility is the entry of device configuration commands to specify any nonstandard devices attached to the system. For example, the diskette magazine drive is standard, therefore it needs no configuration commands entered. Work stations and work station printers require entry of device configuration (device description) commands.

Security

The System/38 security function has three major features:

1. It provides integrity, reducing the possibility of inadvertent destruction or alteration of programs and data by a work station user or someone not trained in data processing.

2. It provides user identification, which allows information that is accessed and displayed to be restricted to specific users.

3. It can be used to prevent disclosure and alteration of sensitive data.

All of these features become increasingly important in the work station environment where physical security is difficult to maintain.

In the work station environment, there may be many untrained personnel 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 generally maintained online and the minute-by-minute operations of the company become quite dependent on the proper functioning of the system. Therefore, integrity becomes much more 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.

Once the system establishes the identity of the specific work station user, the system and the application programs can be tailored for that user. For example, at sign-on an order entry clerk can be presented with choices for performing order entry applications, a credit manager with credit approval requests, and a programmer with the command entry display.

The major difference in the System/38's security function from most previous systems is that it does not simply limit access to the system through some mechanism such as a common password. Instead, each user of the system can have his own user password. 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 contained 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) allows each user to be authorized only the functions necessary to perform his job.

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

The security function can be implemented in a very simple fashion by using the system-supplied user profiles and defaults. Then, as the company requires a higher level of security, the more comprehensive features of security can be used.

Save/Restore

The System/38 machine and program products are designed to the goal of providing maximum system availability and minimum user concern for recovery.

However, user involvement in backup and recovery considerations is still necessary for certain machine, program product, operational, and power failures. System/38 provides a save/restore function to do the traditional backup and reload operations.

The save/restore function:

- Saves objects, or a group of objects in a library, by writing to the diskette magazine drive.
- Restores data that has been written offline.
- Makes the space available for use that was occupied by data after the data was written offline, but keeps the objects description and the offline location.

Information about the saved objects is kept in a save/restore directory. For each object, there is one entry that reflects the most recently saved copy. The directory is indexed by both object name and diskette volume identifier. The save/restore function keeps the directory updated.

Save/restore functions also include directory inquiry and print operations.

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 are able to share the use of the system console. Information displayed at the system console can assist the service representative in faster problem diagnosis.

The following items are available to assist the service representative:

- History file (system operator messages and responses)
- Error recording area (a defined storage area for accumulating machine and program errors)
- System level files (a defined storage area to hold current level information)
- System status files (the status of the system at the time of failure)
- Traces (levels are included for the type of function to be traced)
- Display facilities (allows a dump to the printer or console)
- Modification facilities (allows 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 the usefulness and 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* allows the user to eliminate or minimize the input and output specifications for externally described data files. This provides ease of coding, consistent naming conventions, and improved growth capability. Many programs can now be written without the use or with only minimal use of input and output specifications.

- *Full procedural file specification* allows the user to process the same file in both a random and sequential manner in the same program (by use of explicit input/output specifications). This allows many new application approaches to be solved in a straightforward manner.

- *Explicit input/output calculations* provide enchancements to the following.

  Existing file processing operation codes:
  - READ (can now read a specific record type or format)
  - CHAIN (can now chain to a specific record type or format)
  - SETLL (can now get an indication of an equal condition)

  New file processing operation codes:
  - READE (read equal for generic group processing)
  - READP (read previous for reading backwards along an access path)
  - WRITE (write a new record without the use of output specifications)
  - UPDAT (update of an existing record without the use of output specifications)
  - DELET (deletion of an existing record)
  - SETGT (positions the file so the specified key is greater than the previous record key)

  New control operation codes:
  - OPEN (programmer control of when to open a file)
  - CLOSE (programmer control of when to close a file)
  - FEOD (force end of data to help control multiple logical files on a single device)

- *Program structure* allows the user to 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)

- *The capability to allow any RPG III program to call other programs.* This allows parameters to be passed during the call and return of programs. Programs can be either RPG III 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)

  A new indicator (RT) also exists which enables programs written using the RPG III program cycle to return to the caller.

- *Data structures* allow the redefinition (one or more times) of a storage area and the processing of 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 MOVEL operations. An externally described data format can also be used to describe a data structure. Specific support exists for a job communication area (done by a data area in the QTEMP library).

- *Multiple occurrence data structures* allow many advanced applications to be programmed in a straightforward manner. This eliminates the need for many arrays and provides a more logical approach to some programmer requirements.

- *Compare and branch operations (CABXX)* allow the user to do a simple compare and branch in one operation and eliminate the need for the setting and testing of many indicators. The xx portion of the CABXX operation code may be GT, LT, EQ, NE, GE, LE, and ƀƀ (unconditional).

- *Short form of calculations* allow the use of a simple ADD, SUB, MULT, and DIV operation where one of the factors can be implicitly specified. For example:

FACTOR1	OPCODE	FACTOR2		
QTY	ADD	TOTQTY	TOTQTY	Traditional statement
	ADD	QTY	TOTQTY	New option

This eliminates unnecessary coding and provides more readable programs.

- *EXCPT to a label* allows simplified control of one or more exception output lines.

- *Simplified subroutine specification* eliminates the requirement to code SR in positions 7-8 of every subroutine statement. This eliminates coding and many diagnostics caused by careless errors.

- *Indicators as data* allows the user additional control of indicator status. This is helpful when communicating to other programs. In addition, the indicator array (*IN) containing indicators 01-99 eliminates coding and provides a more straightforward approach to many program processing requirements, such as the ability to set on or off a block of indicators.

- *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 provide an RPG III solution for the users who want a structured programming approach.

- *IF/ELSE operations* allow the execution of a series of RPG III operations without the use of branching or indicator control. The IFXX operation allows the use of modifiers in the same manner as 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* allows the RPG III user to control the exception/errors which can be raised by data management, and system and machine functions. This provides the user with options so that the progam can continue operation even though serious exception/errors have occurred.

- *TESTN operation code* allows the programmer to check alphabetic fields to ensure they contain valid numeric data. This allows better validity checking in a more simple manner.

- *DEFN operation code* allows for the definition of fields based on the attributes of another field. This provides more flexibility in the way programs are written so that fewer source code changes and considerations are required during program creation and maintenance.

- *Improved use of U1-U8 indicators* allows the U1-U8 indicators to be set on or off during the program and 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* allows the user to specify the output of a field based upon a relative postion from the previously described field. This allows the user to more easily define simple output records 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* allows the use of resulting indicators on MOVE and MOVEL. This eliminates the need for additional operations to check for blank, zero, or plus/minus conditions.

- *Figurative constants* allow the use of RPG III defined names to eliminate initialization statements and provide a solution for 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* allows the program to retrieve the current time or both the date and time from the system timer. This allows the time stamping of transactions and other functions.

- *SHTDN operation code* allows the program to test an external event which requests the shutdown of the program.

- *SORTA operation code* allows an array to be sorted in either ascending or descending sequence.

- *KLIST/KFLD operation codes* allow the user to specify the description of a key field when it is made up of multiple fields.

- *Special file with PLIST* operation code allows the user to pass parameters when the device is SPECIAL.

- *User-defined edit codes* allow the RPG III programmer to define five edit codes, to handle unique editing.

- *Currency symbol* allows the specification of symbols other than $ for use in edit codes and edit words.

- *Dynamic space/skip function* allows the program to 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* allows the user, at the end of the program, to write a table/array back to the same file it was read from.

- *BITON/BITOF* of arrays allows the use of bit settings on whole arrays.

New Compilation Listing Functions

- *Cross reference list* allows easy access to determine where fields and indicators are used and modified. This simplifies corrections and improves the documentation of the program.

- *Text description from data description specifications* allows the programmer to see any text descriptions provided for fields and indicators specified in the data description specifications. These appear on the listing to help document the program.

- *Resulting indicator usage* allows a quick determination of which resulting indicator is being defined (the first, second, or third positions). This eliminates confusion and results in better program documentation.

- *Consecutive position rule* allows the programmer to 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 allows the programmer to go directly from a diagnostic which discusses a source statement in error to the specific statement using the source entry utility.

- *Source diagnostic error* notation allows easy pinpointing of errors by the use of position 1 of the compiler listing. Position 1 will contain an asterisk for any error found. Asterisks will also be placed under each field in error during the listing of the source.

- */EJECT and /SPACE specifications* allow the skipping of lines in the source listing for easier reading of major program functions.

- *Nesting indication* allows the programmer to 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 which produces source output and invokes the RPG III compiler. The automatic formatting functions and /COPY can be used to simplify programming requirements.

Enhanced Debug Facilities

- *Support of the system debug facility* allows online debugging of the RPG III programs without any special source statements. This includes such functions as:
  - Stopping at specific statement numbers or RPG III major functions (for example, the first detail calculation).
  - Displaying and changing of fields, indicators, arrays, etc.
  - Tracing of 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 and allows multiple members for unique source programs.

- A Create RPG Program (CRTRPGPGM) command exists to generate object programs. This includes various options for control of the source listing and security functions.

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 langauge source files.
- *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 aids the user in entering and maintaining source statements for RPG III, control language, data description specifications, and other source formats.

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

The major functions of the Source Entry Utility assist the user in:

- Adding, deleting, and changing source lines of a source file.
- Taking lines from one source file and including them in another.
- Searching a source file for specific character strings.
- Editing fields and performing limited syntactical analysis on fields.
- Performing sequence numbering and control.

Data File Utility (DFU)

The Data File Utility provides functions for data entry, file maintenance, and inquiry against data files.

Some of the major functions of the Data File Utility enable the user to:

- Create Data File Utility definitions by specifying the described data file to be used and taking all the defaults. Therefore data entry, file maintenance, and inquiry applications can be created using the field definitions from the described data and a standard display format provided by the Data File Utility.

- Tailor a definition to use only some of the fields in the file.

- Define Data File Utility applications for files containing multiple record formats (types).

- Create Data File Utility definitions to be used in entering or displaying data to or from a file. 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 or different types.

- Enter data at the same time the previous record is being processed, thereby providing good performance for simple data entry jobs in which minimal editing is performed at the time of data entry (similar to an offline data entry operations).

- Provide an audit trail of data entered in file maintenance operations.

- Provide a simple printed listing of the records in a file.

- Provide for a simple, direct method for retrieving specific data base records and displaying the records in a specified format.

Query Utility

The Query Utility permits the user to sequence, summarize, and tabulate data in a data base file. A report of 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 Query results in the selection of a set of records, and includes the ability to influence the order of the set and to generate summary reports.

During the Query application definition phase the user 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
- Printer line and page opening
- Arrangement of fields on a report

CONVERSION REFORMAT UTILITY

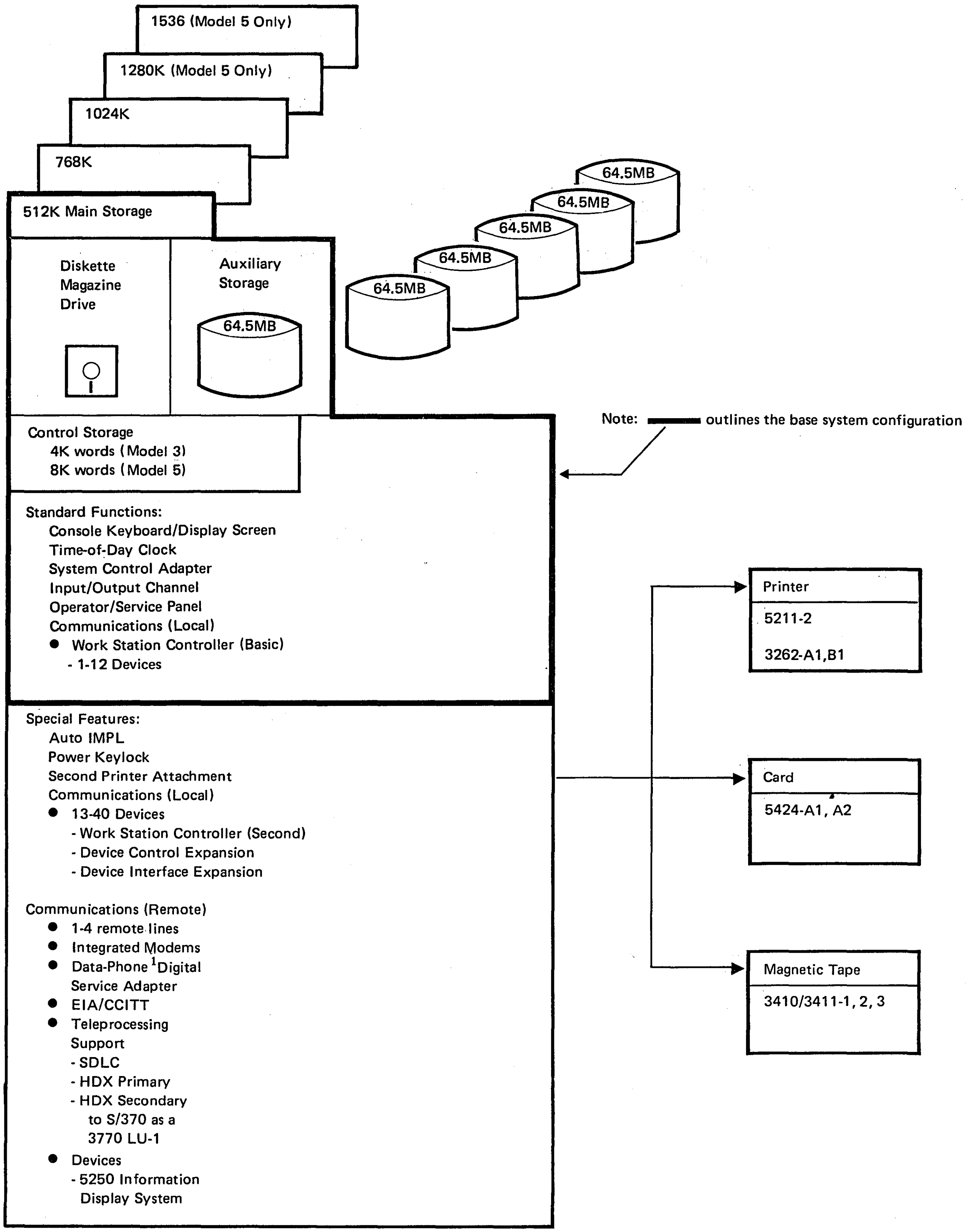
The Conversion Reformat Utility operates upon data from a data base file or a device file to perform sort, merge, and copy operations. The Conversion Reformat Utility allows the user to:

- 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 or device file

Chapter 3. Machine Summary

System/38 is available in two 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.



Control Storage
4K words (Model 3)
8K words (Model 5)

Standard Functions:
Console Keyboard/Display Screen
Time-of-Day Clock
System Control Adapter
Input/Output Channel
Operator/Service Panel
Communications (Local)
- Work Station Controller (Basic)
  - 1-12 Devices

Special Features:
Auto IMPL
Power Keylock
Second Printer Attachment
Communications (Local)
- 13-40 Devices
  - Work Station Controller (Second)
  - Device Control Expansion
  - Device Interface Expansion

Communications (Remote)
- 1-4 remote lines
- Integrated Modems
- Data-Phone [1]Digital Service Adapter
- EIA/CCITT
- Teleprocessing Support
  - SDLC
  - HDX Primary
  - HDX Secondary to S/370 as a 3770 LU-1
- Devices
  - 5250 Information Display System

Note: ▬ outlines the base system configuration

Printer
5211-2
3262-A1,B1

Card
5424-A1, A2

Magnetic Tape
3410/3411-1, 2, 3

Trademark of American Telephone & Telegraph Co.

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 system unit consist of the:

- 5381 Processing Unit and its related control storage
- Virtual address translator
- Input/output channel
- Main storage

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 previously found in system programs.

The object-oriented architecture is fundamental to the overall design of the powerful functions provided by System/38. An object is storage used to implement a construct such as a program, process, or data base file. Although the machine user can control object storage performance characteristics, object storage is managed entirely by the machine using virtual addressing techniques.

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

The access path to objects is machine controlled. 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 obtaining of addressability to unauthorized objects or virtual storage used by the microcode. These features provide greater data integrity and security than available on previous systems.

Each user of the machine executes as an independent process. The machine resources (processor, storage, devices) shared by processes are also managed by the machine. Inter-process 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 microcode before execution to achieve greater efficiency. Program variable attributes allow the machine to 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 intricacies of the channel, communication networks, and asynchronous device operations are handled by the machine.

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.

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

Processing Unit

The 5381 processing unit 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.

In performing its required functions, the processing unit uses internal storage called control storage. Control Storage provides random-access storage for the powerful, high-level, 32-bit word, microinstruction set (microcode) used by the processing unit. Many of the functions formerly coded in system supervisors are implemented in the System/38 microcode, resulting in a more efficient system.

The 5381 processing unit is available in two models (3 and 5). The Model 5 can achieve approximately two times the internal performance level of the Model 3. The Model 3 has 4K words of control storage, while the Model 5 has 8K words.

Storage Management

Main storage is not divided into partitions. Therefore, most operating and design considerations for program size and for concurrent execution of programs are eliminated.

System/38 storage management offers such advantages as:

- Maintaining a directory of locations of all objects (such as files and programs) in the system. The programmer no longer has to consider specific file location, disk volume, drive, or space requirements in the design of his programs.

- Allocating storage space as it is required. Therefore, 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, that space is automatically used by the system without requiring changes to existing programs or control language commands.

- Treating all of auxiliary storage as a part of one continuous volume of space. This permits any file to 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 virtual address translator performs rapid conversion of addresses between auxiliary storage and main storage, thereby supporting an efficient storage management function.

Internal Input/Output Channel

The high-speed, internal input/output channel, with an aggregate transfer rate of up to 2.5 megabytes per second, connects the processing unit, main storage, and the input/output devices. The channel permits the overlapping of input/output operations with processing.

Main Storage Capacity

Main storage capacity provides the system with directly addressable storage for data and programs. The basic main storage module consists of 512K (524 288) bytes of storage and is equipped with error correction code hardware to automatically detect all single and double bit errors and automatically correct all single bit errors. Main storage capacities available with the 5381 Processing Unit Models 3 and 5 are:

- 512K (524 288) bytes
- 768K (786 432) bytes
- 1024K (1 048 576) bytes
- 1280K (1 310 720) bytes (Model 5 only)
- 1536K (1 572 864) bytes (Model 5 only)

The main storage input/output channel to the processing unit is four bytes wide. The average main storage cycle time is approximately 1100 nanoseconds for the Model 3, while main storage in the Model 5 performs at 600 nanoseconds.

Time-of-Day Clock

The time-of-day clock is a timer that runs continuously whenever system power is on. The clock is not affected by system status (stopped or running). Functions are supplied to the user to set the clock and perform actions, such as read the clock.

Automatic Initial Microprogram Load

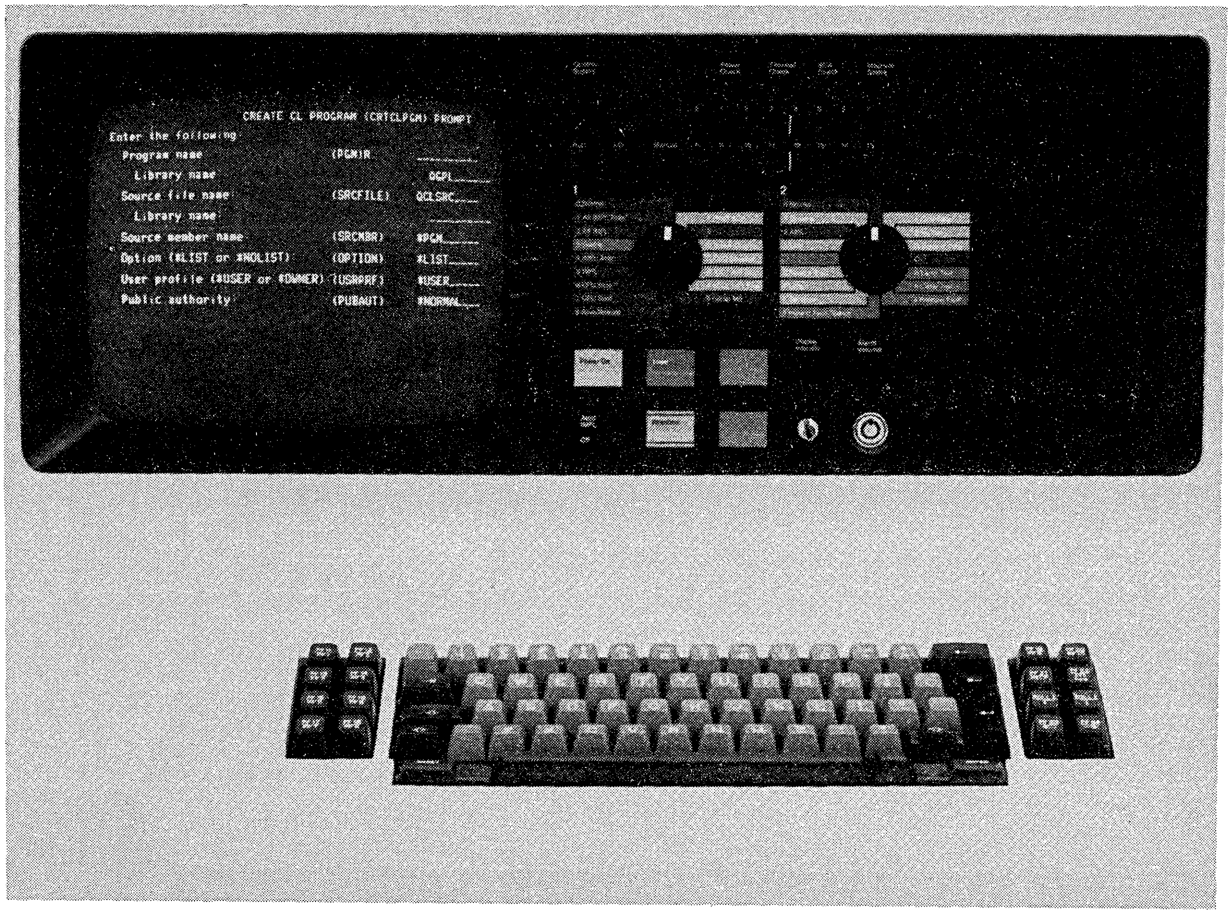
If commercial power is interrupted, this optional feature automatically initiates a power-on sequence after commercial power is restored. Once the Control Program Facility has been loaded into the system, this feature allows a user-provided recovery program to resume operations. (The system operator controls this feature through a switch.)

Power Keylock

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

SYSTEM CONSOLE

The System/38 console (Figure 3-2) includes a keyboard and a display screen. The console is used primarily for interactive communication between the system operator and programs (Control Program Facility and user programs) executing in the system, and between the system operator and work station users.



```
                    CREATE CL PROGRAM (CRTCLPGM) PROMPT
Enter the following:
  Program name                        (PGM)R      ________
    Library name                                   QGPL____
  Source file name                    (SRCFILE)   QCLSRC__
    Library name                                   ________
  Source member name                  (SRCMBR)    *PGM____
  Option (*LIST or *NOLIST)           (OPTION)    *LIST___
  User profile (*USER or *OWNER)      (USRPRF)    *USER___
  Public authority                    (PUBAUT)    *NORMAL_
```

Figure 3-2. System Console

System Console Display Screen

The system console display screen (Figure 3-3) 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 intensity of the display can be changed by a control on the operator/service panel.

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

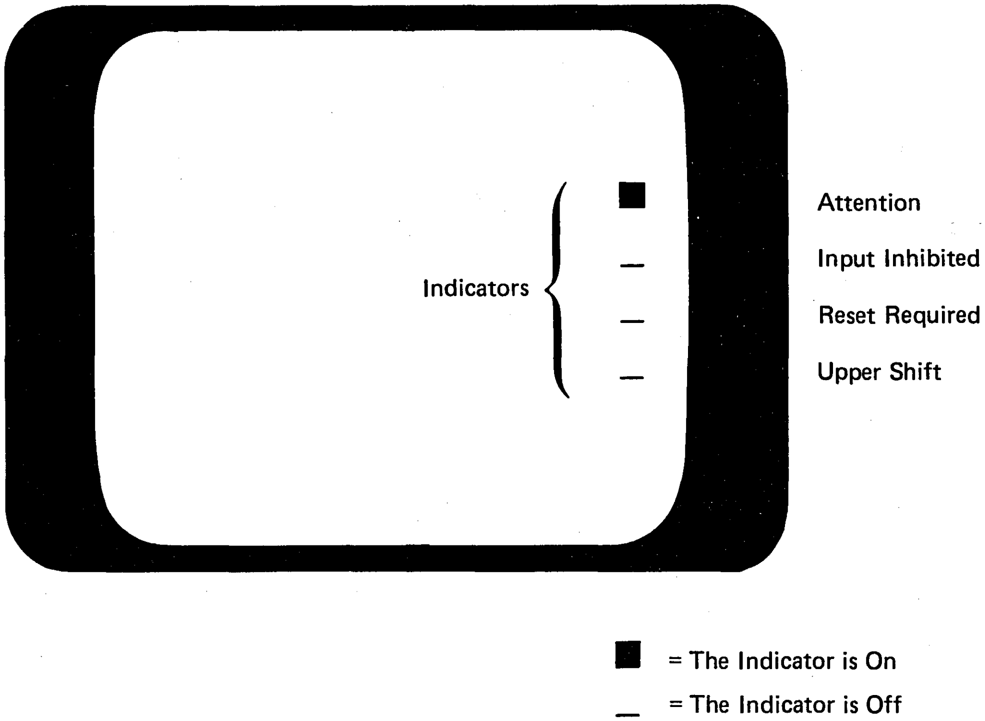


Figure 3-3. System Console Display Screen

The system console can be used:

- For system operator functions.
- For programmer functions.
- For application functions (similar to a work station).
- For maintenance functions (by the service representative to run tests and to display diagnostic information).

System Console Keyboard

The system console keyboard (Figure 3-4) is used by the system operator 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 (upper and lower case) and numeric data.

- Cursor movement keys to position the cursor on the display screen.

- 24 easily identified command function keys for requesting specific program functions.

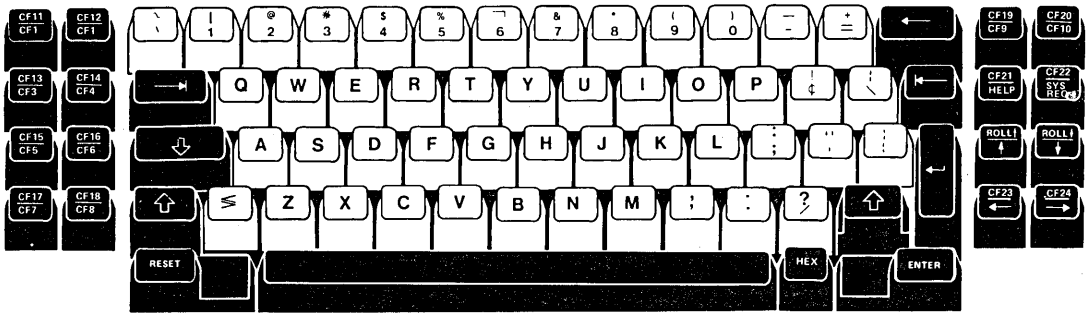


Figure 3-4. System Console Keyboard

System Console Operator/Service Panel

The operator/service panel (Figure 3-5) 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. Additional switches may be used by the service representative to control the operations of the system.

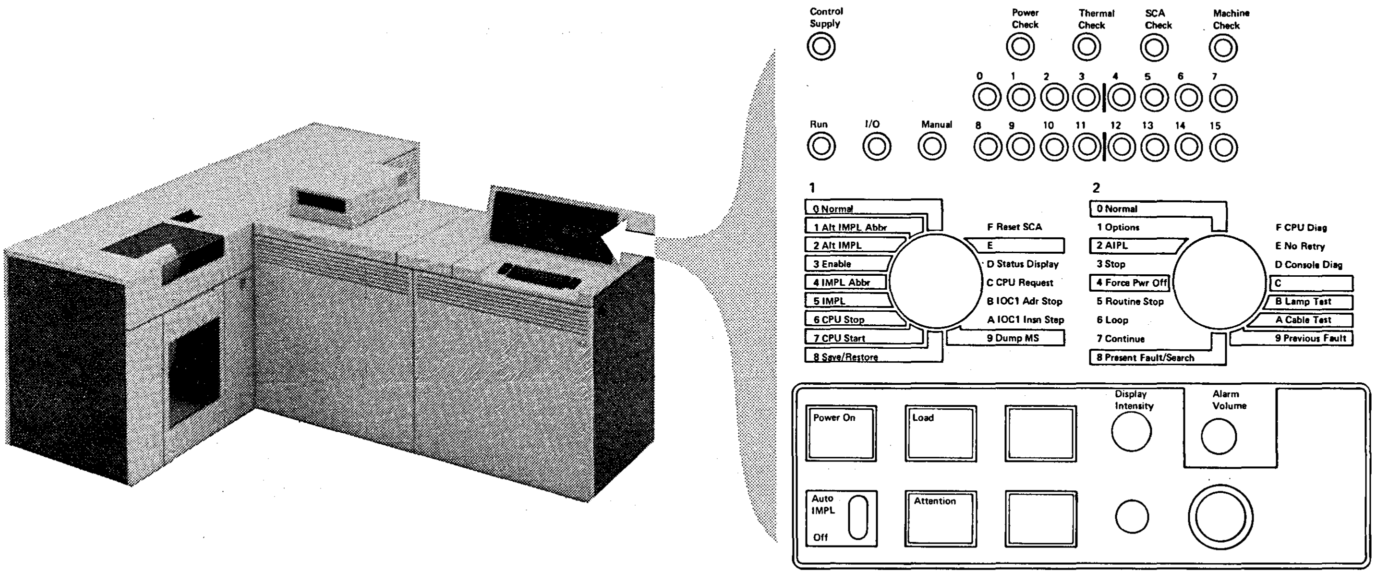


Figure 3-5. System Console Operator/Service Panel

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, and the system console).

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 or supplying a synchronous pulse when the system encounters a specified address.

AUXILIARY STORAGE

Online auxiliary storage is supplied by the *data storage function.* The data storage function provides the System/38 users with auxiliary storage (magnetic disk) starting with a minimum of 64.5 megabytes and increasing in 64.5 megabyte increments (up to a total of 387 megabytes).

Each data storage function disk enclosure supplies 64.5 megabytes of storage media. One to six nonremovable disk enclosures are available.

Disk enclosures 1 and 2 are packaged in the base enclosure unit. Disk enclosures 3, 4, 5, and 6 are in the expansion enclosure unit (Figure 3-6).

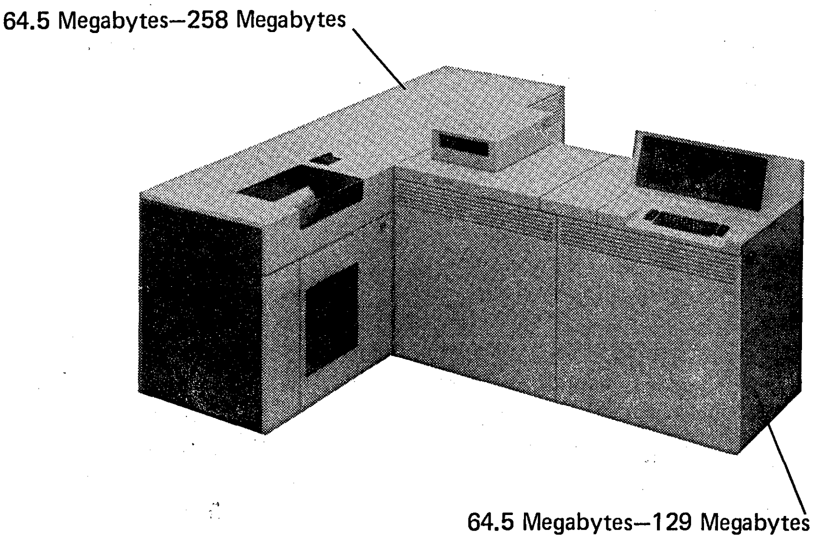


Figure 3-6. Auxiliary Storage Disk Enclosure Locations

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

- Access times: 23.5 milliseconds (average)
- Data transfer rate: Up to 1 031 000 bytes per second
- Specification of the disks

  Number of sectors: Each track has 33 sectors, 0 through 32. One sector, normally unassigned, is for alternate use. That is, if a sector in a track fails, its use is assigned to the same track in the alternate sector.

  Size of sector: Each sector can accommodate 512 bytes of data.
- Capacity: 64 520 192 bytes

DISKETTE MAGAZINE DRIVE

The diskette magazine drive (Figure 3-7) is standard on all models of System/38 and 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, program changes, and new program releases from the IBM Program Information Department (PID).

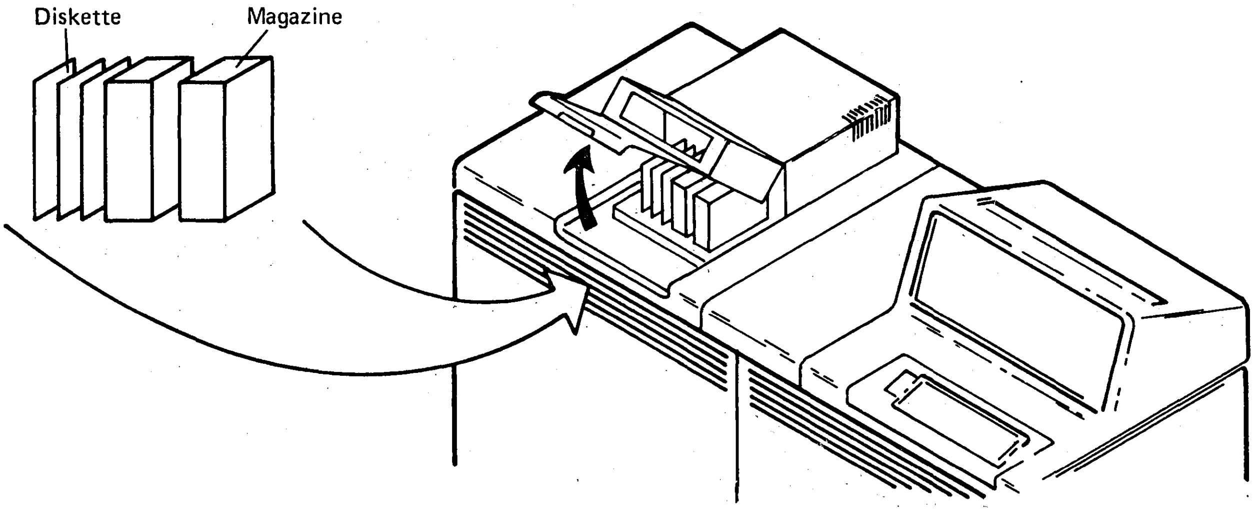


Figure 3-7. Diskette Magazine Drive

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 permanently enclosed in a protective jacket. Each diskette is removable from the diskette drive.

The following are the diskette types and their formats:

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

* Only supported at the machine interface level.

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. If magazine has been replaced, the processing can continue again with this magazine.

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 magazine concept, diskette handling, labeling, and storage are simplified.

- Random access to any data set on any diskette.

- Multifunction device operations.
  - Save/restore
  - Spooling
  - Copy
  - RPG III file operations

PRINTERS

5211 Printer; 3262 Printer

System/38 supports one model of the 5211 Printer, and two models of the 3262 Printer, as the system's printer.

Type	Model	Rated Speed* (Lines per Minute)
5211	2	300
3262	A1, B1	650

* Rated speeds are based on a 48-character set print belt.

Note: The A model attaches directly to the system unit.

Both types of printers (Figure 3-8) have improved setup features and quiet operation. The quiet operation is achieved by feeding paper completely within the covers. Forms control, as well as selection of 6 or 8 lines per inch, is programmable.

Both the 5211 and the 3262 printer are line printers; that is, they print one complete line at a time, up to 132 characters per line. Speeds of the 5211 and the 3262 will 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 from 3 inches to 14 inches long.

The outward physical appearance of the 3262 printer is identical to the 5211 printer (Figure 3-8).

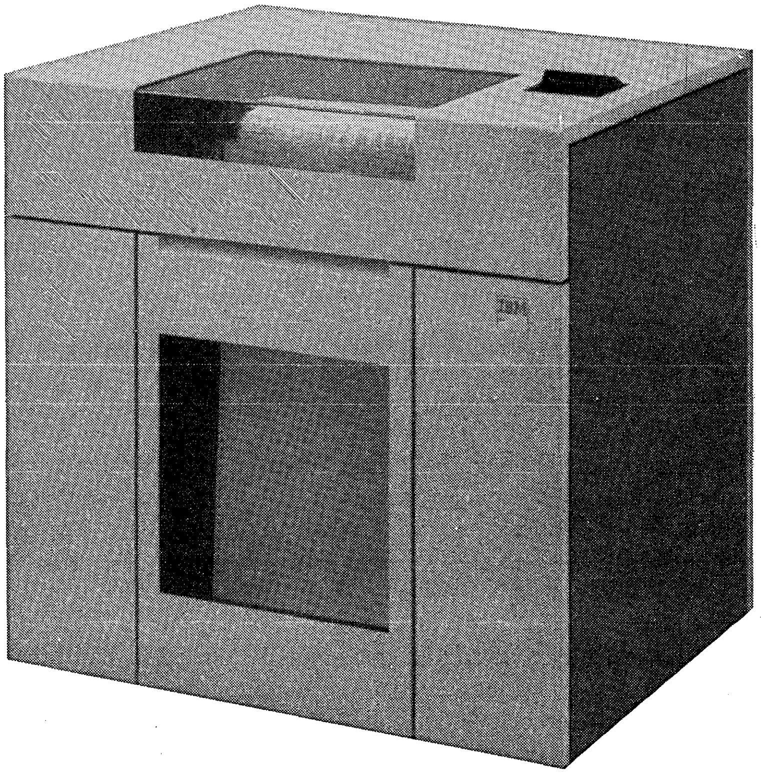


Figure 3-8. 5211 Printer; 3262 Printer

System/38 can attach up to two system printers. The various models of the 5211 and the 3262 can be attached in any combination, with one limitation; System/38 will not support two directly attached A models.

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

Specialized print belts that may yield higher speeds are available for the 5211 and 3262 printers.

5211 Printer

A universal character set buffer of 192 positions in the 5211 Attachment allows use of graphic sets of up to 192 characters. Print speeds 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 offering the following nominal rated speeds:

	Speed (Lines per Minute)
48-character set	300
64-character set	235
96-character set	164

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

Model	**Speed (Lines per Minute)**
5211-2	355

- The optimized numeric belt contains 42 graphics (A-Z, 0-9, six special characters) and is optimized to the printing of the numbers, the period, and the comma. When printing single spaced numeric information, the following speed can be reached:

Model	**Speed (Lines per Minute)**
5211-2	395

When printing the graphics (A-Z), with the 42-character set belt, the following rated speed can be reached:

Model	**Speed (Lines per Minute)**
5211-2	300

3262 Printer

A universal character set buffer of 288 positions in the 3262 Attachment allows use of graphic sets of up to 288 characters. Standard print belts of 48, 64, and 96 characters are available which offer the following nominal rated speeds:

	Speed (Lines per Minute)
48-character set	650
64-character set	467
96-character set	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 64-character normal print belt.

60-Character System/38 Special Belt

The 60-character special belt enables the user to print the System/38 control language characters.

The 60-character System/38 special belt is available on all models of the 5211 printer and the 3262 printer.

CARD DEVICE

5424 Multi-Function Card Unit

The IBM 5424 Multi-Function Card Unit is offered as an optional feature on System/38.

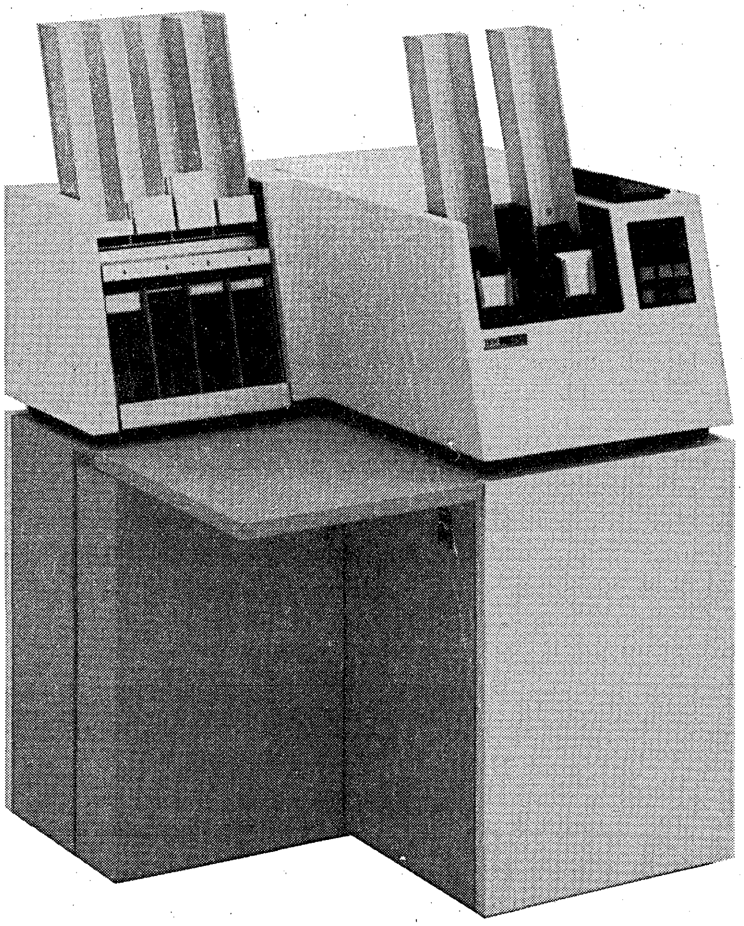


Figure 3-9. IBM 5424 Multi-Function Card Unit

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

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	**Model A2**
Reads 250 cards per minute	Reads 500 cards per minute
Punches 60 cards per minute	Punches 120 cards per minute
Prints 60 cards per minute	Prints 120 cards per minute

MAGNETIC TAPE DEVICE

3410/3411 Magnetic Tape Subsystem

The IBM 3410/3411 Magnetic Tape Subsystem offered as an optional feature on System/38 consists of one 3411 Magnetic Tape Unit and Control (both units are in the same frame). One to three additional 3410 Magnetic Tape units can also be attached.

There are three models of the 3410/3411 subsystem.

- The Model 1 moves tape at 12.5 inches per second resulting in a maximum data rate of 20 000 bytes per second.
- The Model 2 moves tape at 25 inches per second resulting in a maximum data rate of 40 000 bytes per second.
- The 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 Subsystem 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 3411 are table-high units with tape reels installed horizontally rather than vertically. A transparent sliding cover permits easy access to the tape reels.

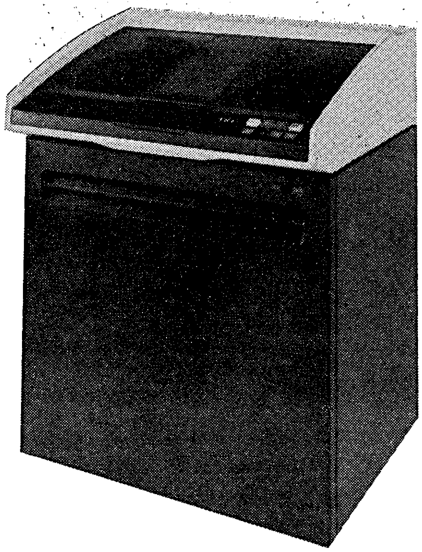


Figure 3-10. IBM 3410 Magnetic Tape Unit

COMMUNICATIONS ADAPTERS

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 3-11): One is through the work station controllers, the other through the communication attachment.

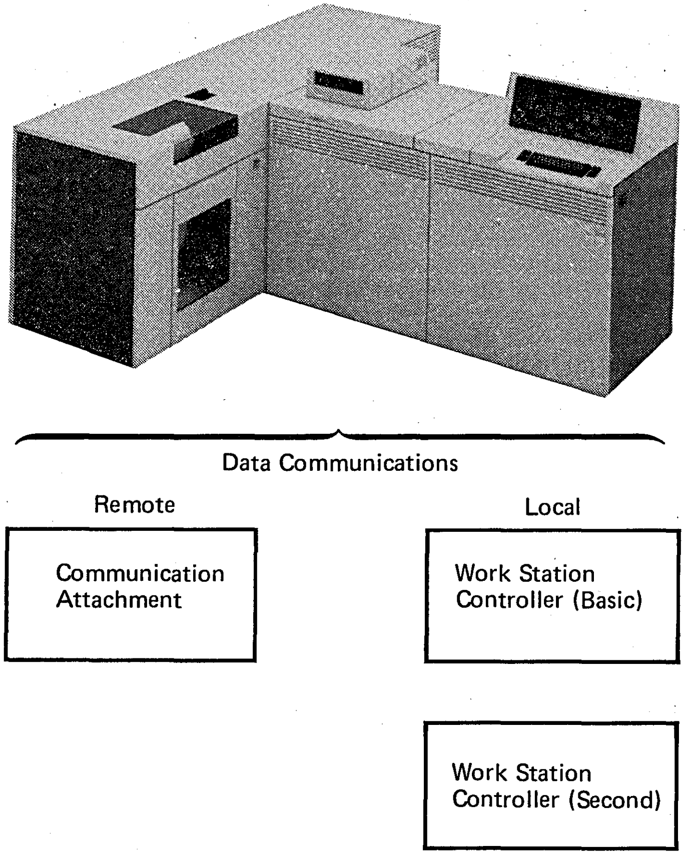

Data Communications
Remote
Local
Communication Attachment
Work Station Controller (Basic)
Work Station Controller (Second)

Figure 3-11. Local and Remote Communication Attachment

Work Station Controllers

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

Up to 40 work stations (the 5251 Display Station Model 1 and 11, the 5252 Dual Display Station Model 1, and the 5256 Printer Models 1, 2, and 3) can be connected directly to System/38 through this attachment method.

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

Work Station Controller (Basic)

This 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 (in a single and multipoint arrangement) of up to 12 work stations (Figure 3-12).

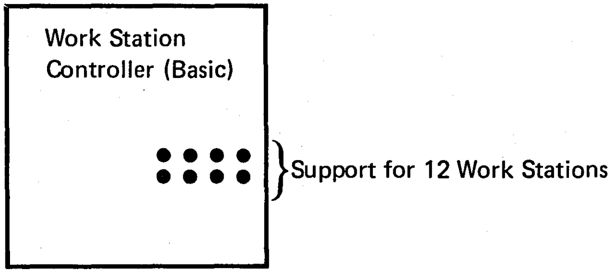


Figure 3-12. Work Station Controller (Basic)

Twinaxial Cable

Twinaxial cable is recommended for attaching the work stations because it allows a maximum length of 1525 meters (5000 feet) and multiple work stations (seven) to be attached to a single cable. Figure 3-13 shows an example of attaching work station on one twinaxial cable. This cabling arrangement requires use of the 5250 Cable Thru feature.

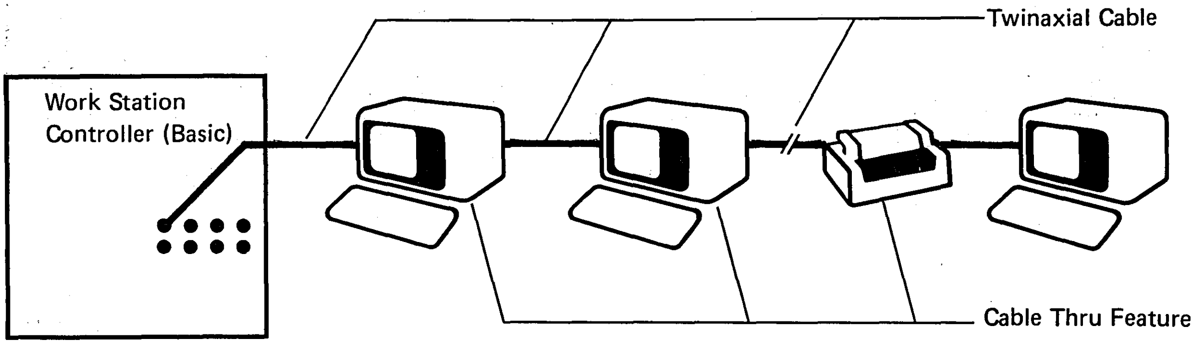


Figure 3-13. Example Using Twinaxial Cable

Coaxial Cable

Those users who now have IBM 3270 Information Display System devices can use the existing one-wire (coaxial), shielded cable when installing IBM 5250 Information Display System Devices. A Twinaxial/Coaxial Adapter is required at both ends of the cable to connect the 5250 devices.

The maximum allowable length of a coaxial cable is 610 meters (2000 feet) and only one work station may be attached to a coaxial cable. A second work station may be attached through the Cable Thru feature. Twinaxial cable must be used between the first and second work stations, and the maximum allowable distance between these two work stations is 30 meters (100 feet). Figure 3-14 shows an example of this attachment method.

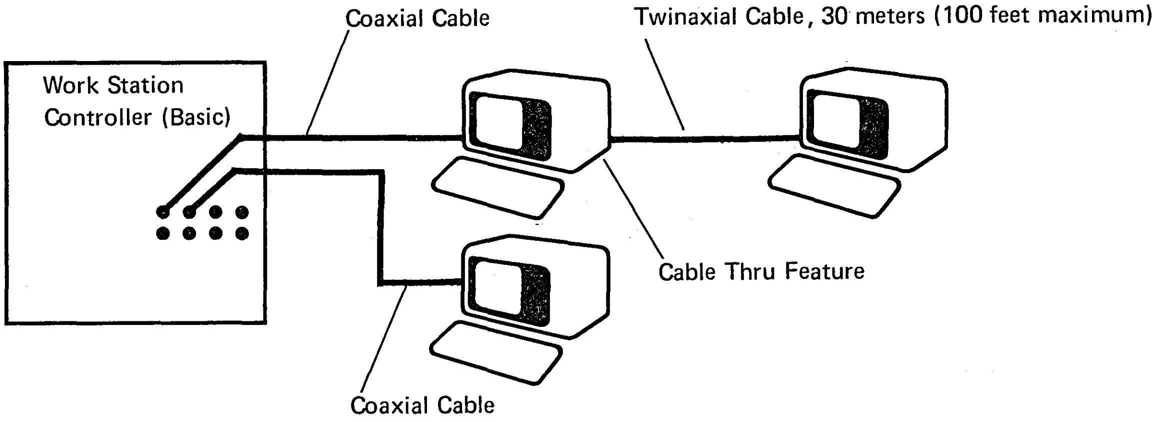

Coaxial Cable
Twinaxial Cable, 30 meters (100 feet maximum)
Work Station Controller (Basic)
Cable Thru Feature
Coaxial Cable

Figure 3-14. Example Using Existing Coaxial Cable

If both devices (work stations) attached by this method were to be IBM 5252 Dual Display stations, the user would effectively have four work stations attached.

Device Control Expansion

This special feature permits the work station controller to 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 3-15).

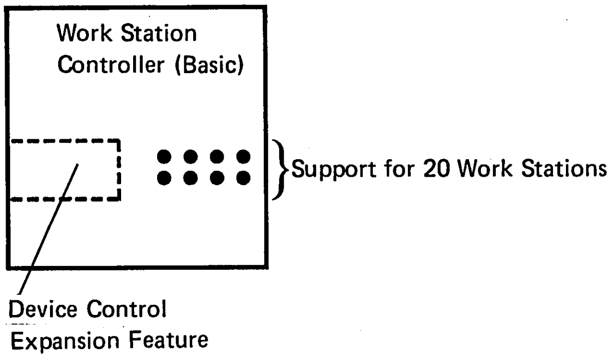


Figure 3-15. Device Control Expansion

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

Device Interface Expansion

This special feature provides the necessary control and eight more ports for the attachment of additional work stations (Figure 3-16).

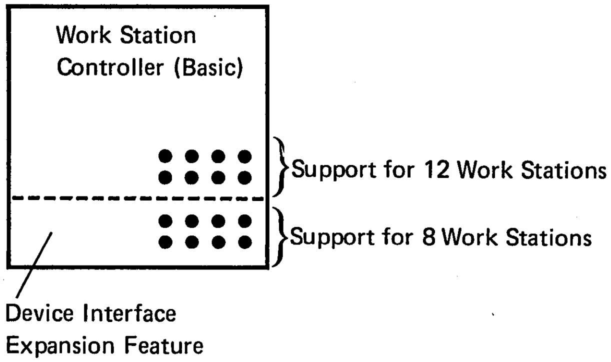


Figure 3-16. Device Interface Expansion

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

Work Station Controller (Second)

This work station controller is a special feature available on all models of System/38. This controller provides support identical to that of the Work Station Controller (Basic). For attachment of additional work stations, see the discussions about the Device Control Expansion and the Device Interface Expansion special features.

Communication Attachment

The communication attachment provides the necessary system control for the direct attachment of up to four remote communications lines. The communications attachment (Figure 3-17) permits the attachment of those devices that require data transmission services (such as private or common-carrier-supplied lines).

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

Communication
Attachment

- Integrated Modems
- Data-Phone Digital Service Adapter
- EIA/CCITT
- Teleprocessing Support
  - SDLC
  - HDX Primary
  - HDX Secondary to S/370 as a 3770 LU-1
- Devices
  - 5250 Information Display System

Figure 3-17. Remote Communication Support

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

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) as a part of 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/VS, OS/VS1, or OS/VS2 (MVS) in System/370 (models 145 to 168 for IMS/VS and models 135 to 168 for CICS/VS and 3031, 3032, and 3033 processors). Communication to any of these operating systems when running under VM/370 is also supported. Communication is via levels of VTAM, ACF/VTAM, 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.

WORK STATIONS

5250 Information Display System

The 5250 Information Display System, which consists of tabletop display stations and printers, is designed for data entry and inquiry applications. These work stations can be set up in areas away from System/38, such as the sales counter, order entry department, and shipping departments.

The 5250 Information Display System offers:

- A 5251 Models 2 and 12 Display Station that provides remote communications with System/38 using SDLC line discipline.

- A 5251 Models 1 and 11 Display Station, 5252 Dual Display Station, and 5256 Printer that can be cable-connected directly to System/38.

- A Cable Thru feature for connecting 5251 Models 1 and 11 Display Stations, 5252 Dual Display Stations, and 5256 Printers in series.

- A Cluster feature that allows the 5251 Models 2 and 12 Display Station to control up to eight 5251 Models 1 and 11 Display Stations, 5252 Dual Display Stations, or 5256 Printers or a combination of all three.

  Note: The 5252 represents two work stations because it has two keyboards.

- The functional capability to display and print uppercase and lowercase alphabetic characters.

- Display highlighting functions, such as blink, column separator, high intensity, nondisplay, reverse image, and underscore.

IBM 5251 Display Station

The following characteristics are common to all models of the 5251 (Figure 3-18):

- A display screen (two sizes are available).
  - Models 1 and 2: 960 characters (12 lines of 80 characters)
  - Models 11 and 12: 1920 characters (24 lines of 80 characters)

- A movable keyboard with several key arrangements.

- A keyboard command key for selecting up to 24 additional, programmable, command functions.

- (Models 2 and 12 only) An internal communications adapter for the remote attachment to a common carrier communications line using the SDLC line discipline with:
  - Transmission speeds of up to 9600 bits per second.
  - Two cluster features that allow the direct attachment of up to eight 5251 Models 1 and 11 Display Stations, 5252 Dual Display Stations, or 5256 Printers or a combination of all three.

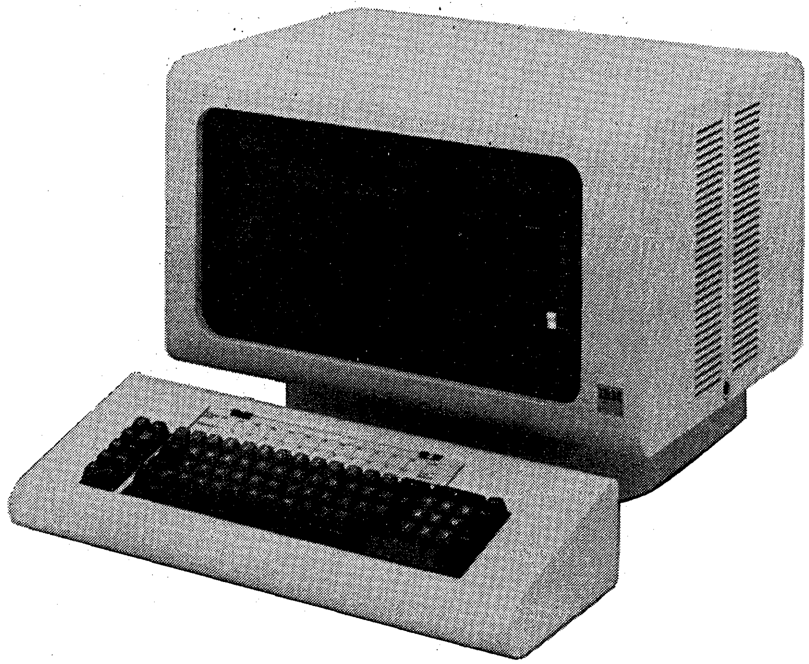


Figure 3-18. 5251 Display Station

5252 Dual Display Station

The 5252 Dual Display station (Figure 3-19) is designed for use by two operators in high-production data entry applications or where multiple display stations are required at a location. The 5252 attaches directly to System/38 or to a 5251 Model 2 or 12 Display Station.

A dual display station has two movable keyboards that are connected by a short cable to a common display unit. This display unit is divided in half logically and optically into two independent work stations, which allows each operator to view a separate 960-character display with up to 12 lines of 80 characters each. The 5252 provides the same display control and highlighting of individual fields as the 5251 and also uses the same keyboards.

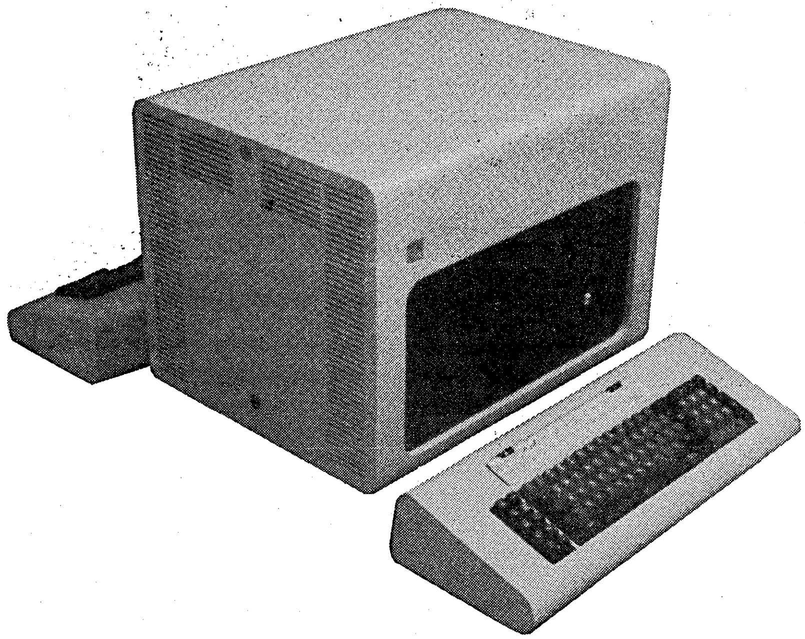


Figure 3-19. 5252 Dual Display Station

5256 Printer

The 5256 Printer (Figure 3-20) has these characteristics:

- A printing speed of 40, 80, or 120 characters per second.
- Bidirectional, serial matrix printing.
- Full look-ahead capability for most efficient throughput.
- A maximum print line of 132 characters.
- The capability to use individual or continuous forms.
- Print position spacing of 10 characters per inch.
- Selectable line spacing of 6 or 8 lines per inch.
- The capability to print on continuous forms that have an original and five carbons.
- An Audible Alarm feature.
- Forms control to control spacing and skipping of the forms that have sprocket feed holes.

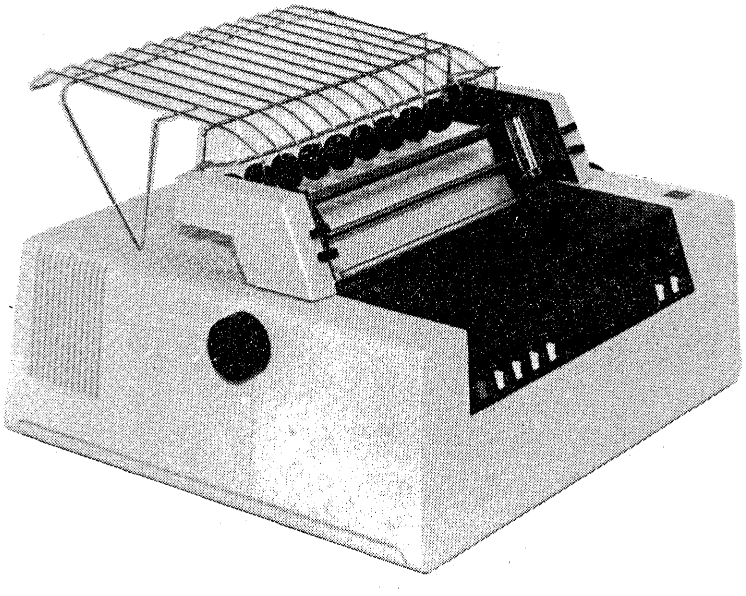
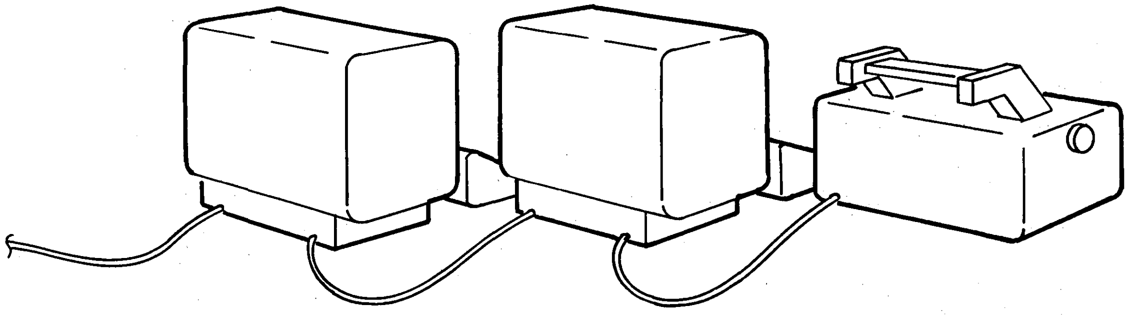


Figure 3-20. 5256 Printer

The Cable Thru Feature

The Cable Thru feature provides the capability of attaching multiple display stations or printers to a single cable (Figure 3-21). A maximum length of 1525 meters (5000 feet) of cable is allowed between System/38 or a 5251 Model 2 or 12 and the last work station on the cable.



To System/38 or a 5251 Model 2 or 12

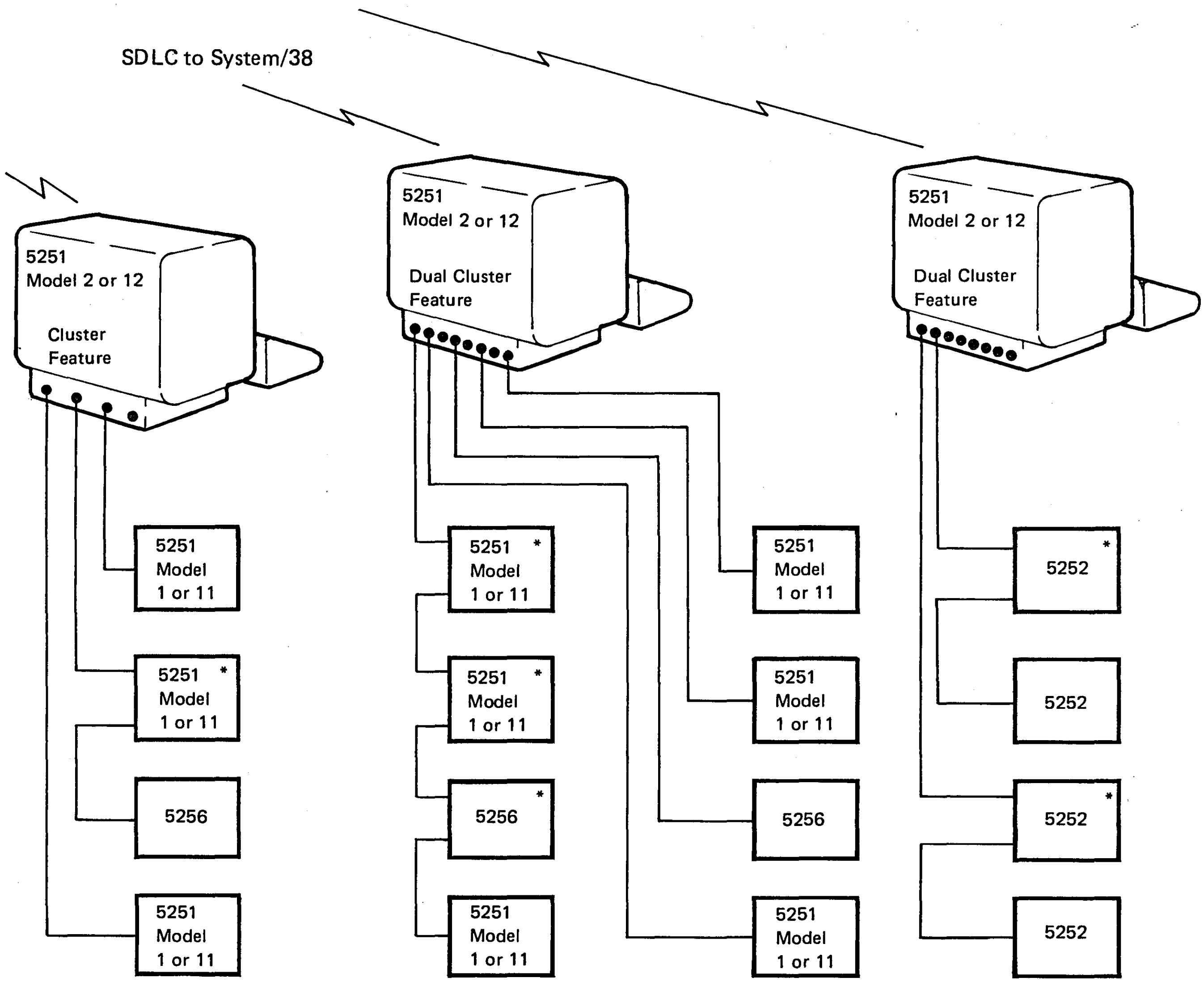
Figure 3-21. Cable Thru Feature

Cluster Features

Two mutually-exclusive cluster features (Cluster or Dual Cluster) are available for the direct attachment of 5256 Printers, 5252 Dual Display Stations, or 5251 Model 1 or 11 Display Stations, to the Model 2 or 12.

The Cluster feature provides four cable connections and allows the attachment of up to four work stations. The Dual Cluster feature provides eight cable connections and allows the attachment of up to eight work stations.

Figure 3-22 shows three 5251 Model 2 or 12 Display Stations in cluster configurations: one display station has the Cluster feature and the other two display stations have the Dual Cluster feature. Each 5251 Model 2 or 12 is controlling the maximum number of directly attached work stations. Note that the 5252 represents two work stations because it has two keyboards.



*Cable Thru feature

Figure 3-22. Cluster Configurations

Keyboards

There are three different keyboard arrangements available: the typewriter-like, data entry, and data entry with proof arrangement.

The typewriter-like keyboard, shown in Figure 3-23, has the basic keyboard arrangement found on a standard typewriter with an extra set of numeric keys arranged like a calculator keyboard.

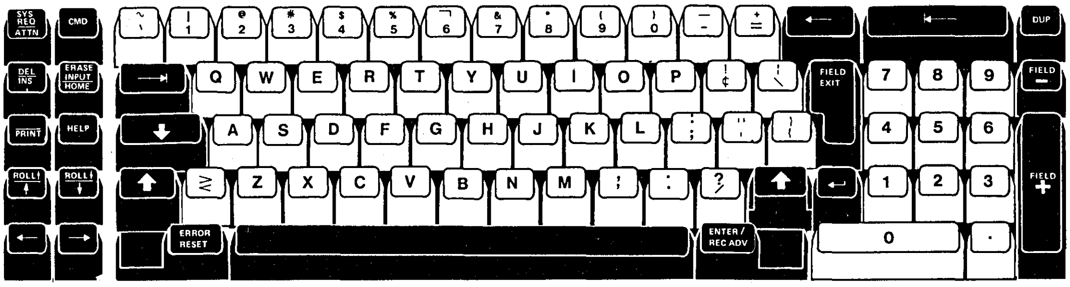


Figure 3-23. Typewriter-like Keyboard

The data entry keyboard, shown in Figure 3-24, 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.

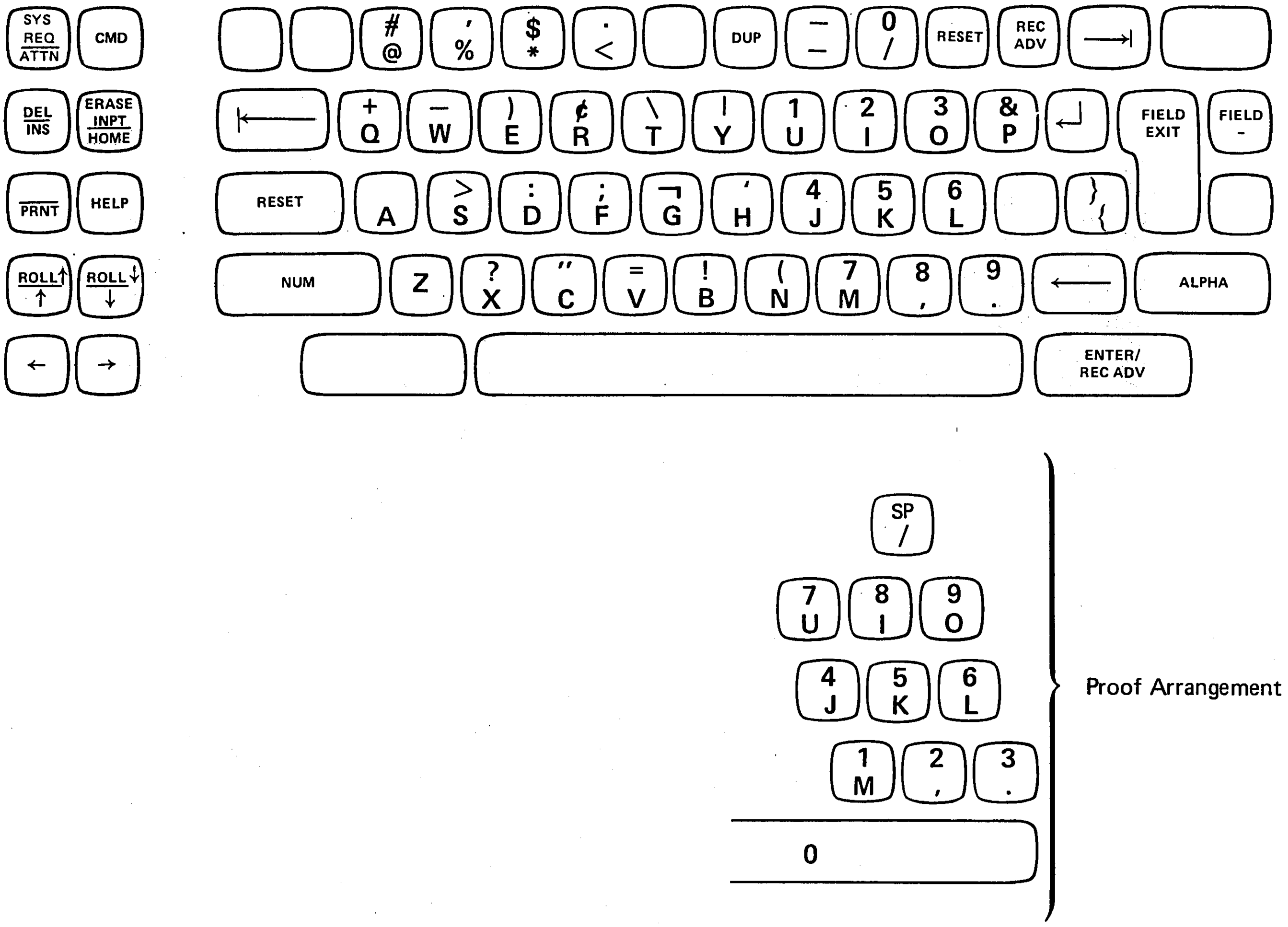


Figure 3-24. Data Entry Keyboard

Magnetic Stripe Reader

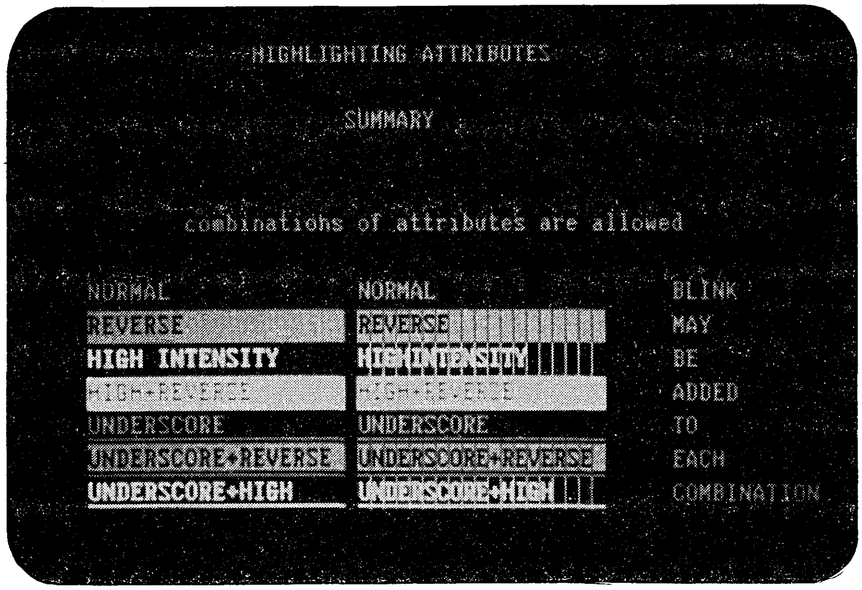
The Magnetic Stripe Reader feature is supported by System/38 on the 5251 Models 2 and 12 (the 5252 and the 5251 Models 1 and 11 are supported through the Cluster feature) and provides the capability to read numeric encoded information from a magnetic stripe. The stripe may be used on a wide range of credit cards, operator identification badges, and other magnetic encoded documents. The magnetic stripe may be encoded with up to 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. The Magnetic Stripe Reader cannot be installed on a 5250 Information Display System connected to the work station controller.

Selector Light Pen

The Selector Light Pen feature is supported by System/38 on the 5251 Models 2 and 12 (the 5251 Models 1 and 11 are supported through the Cluster feature) and is a pen-like device that permits the operator to select fields of data from the display screen for system input.

Display Highlighting Functions

Figure 3-25 summarizes the highlighting functions available on the 5251 and 5252 display screens.



```
                HIGHLIGHTING ATTRIBUTES

                        SUMMARY


        combinations of attributes are allowed

NORMAL                NORMAL                  BLINK
REVERSE               REVERSE                 MAY
HIGH INTENSITY        HIGH INTENSITY          BE
HIGH+REVERSE          HIGH+REVERSE            ADDED
UNDERSCORE            UNDERSCORE              TO
UNDERSCORE+REVERSE    UNDERSCORE+REVERSE      EACH
UNDERSCORE+HIGH       UNDERSCORE+HIGH         COMBINATION
```

Figure 3-25. Highlighting Function

Blink: Causes all characters in the field to blink repeatedly. The entire field is blinked at the rate of approximately one and one-half times per second.

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

High Intensity: Causes the characters in a field to be displayed at a brighter intensity than other data on the screen. The brightness of high intensity fields is adjustable with the contrast control on the display station control panel.

Nondisplay: Allows data to be entered without it being displayed. This attribute is beneficial in applications where there are security restrictions. An identification can be entered in nondisplay fields to prevent unauthorized viewing.

Reverse Image: Causes the characters in the field to be displayed as either dark characters on a light background or light characters on a dark background.

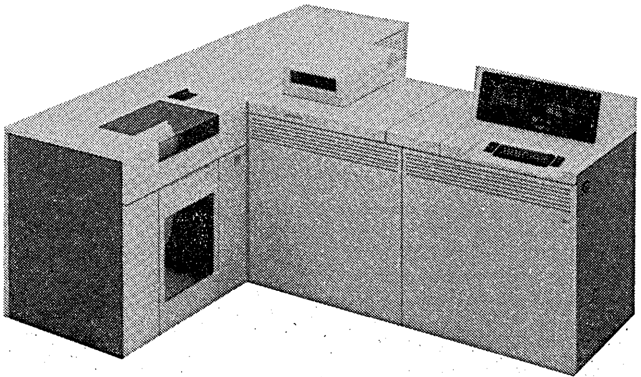
The preceding display highlighting functions, plus blink cursor, can be used to alert the work station user that an error occurred or that a specified action must be performed.

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

Chapter 4. Compatibility/Conversion

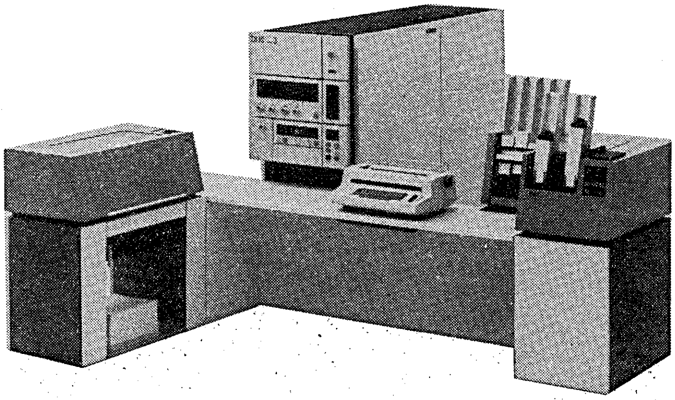
IBM System/38

- RPG III
- Control Language
- Control Language Programs
- Diskette Support
- 96-Column Card Support



IBM System/3

- RPG II
- Operation Control Language
- Procedures
- Diskette Support
- 96-Column Card Support



COMPATIBILITY WITH SYSTEM/3

Even with the major new technological improvement it represents, System/38 can still appear very much like the System/3 to the RPG programmer, if required. He still writes his application in RPG (substantially enhanced and simplified–RPG III) using very similar file structures.

With minor source changes and recompilation, existing System/3 batch programs will execute on System/38. Data files can be moved from System/3 to System/38. This compatibility allows the user to move working application programs to System/38 which can immediately use data base data management functions. Also, new applications can be developed which also use this data in the data base.

System/38 includes most of the utility functions used in System/3 but the need for and the effect of the functions is different.

The System/38 RPG III language is an enhanced extension of the System/3 RPG II language. The data base data management function of System/38 offers all the access methods available on the System/3 Models 8, 10, 12, or 15. The described data enforces field length and type definitions that used to be 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. A field is defined once to the system and use of that field in a program then simply requires specifying the field name.

With System/38, communications lines use SDLC (synchronous data link control). When 5250 devices are installed on System/38, existing, directly attached cables installed for either 3270 devices (on a System/3) or 5250 devices (on a System/34) can be used.

System/38 also supports reading and writing diskettes in the same format used on System/3 and the reading and punching of 96-column cards.

CONVERSION FROM SYSTEM/3

A System/38 conversion utility program product is available to assist the System/3 user in converting to System/38 (this program executes on the user's System/3).

The System/3 Batch Conversion Utilities is a one-time charge program product that automates the actual details of conversion while enabling the user to be in complete control of the overall conversion process.

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

- Converting RPG II source programs to System/38 RPG III source programs.

- Converting 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.

The actual conversion process is under user control. The user may elect to allow the utility to automatically convert everything that needs converting. On the other hand, he may intervene and selectively choose what operational control language and applications are to participate in the conversion process.

Because corrections may be necessary during conversion and because the user may want to alter the conversion output, a comprehensive maintenance service package is included. Through the use of maintenance service, the user can resolve and correct statements that need special attention or modify the conversion process so the output meets his particular needs.

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

Besides performing the actual conversion, the System/3 Batch Conversion Utilities provide for the generation of the interchange media which will be used to move the converted material from the System/3 to System/38. Again, the user controls what interchange media fits his system and if he wants to convert his system in steps or do the whole conversion at once.

Figure 4-1 shows an overview of the System/3 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 which specify the required access paths. Almost all of the 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 conversion, a Conversion Reformat Utility program product (that executes on System/38) is available which creates a TAG file or a reformatted physical file. The reformat utility duplicates all of the functions of the System/3 sort, except summary sort. In addition, the reformat utility uses System/3 sort specifications.

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

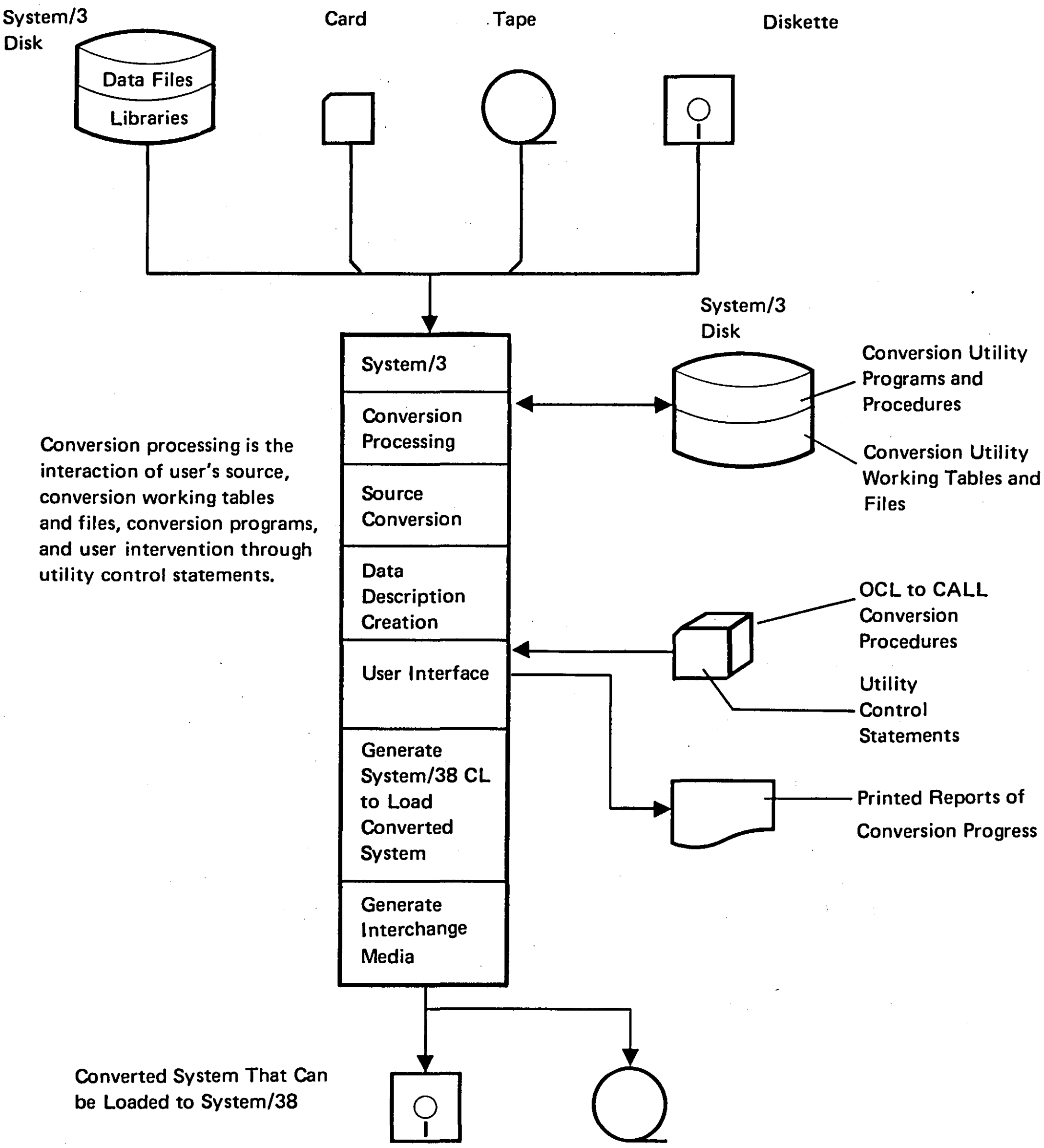

System/3
Disk
Data Files
Libraries
Card
Tape
Diskette
System/3
Conversion Processing
Source Conversion
Data Description Creation
User Interface
Generate System/38 CL to Load Converted System
Generate Interchange Media
System/3
Disk
Conversion Utility Programs and Procedures
Conversion Utility Working Tables and Files
OCL to CALL Conversion Procedures
Utility Control Statements
Printed Reports of Conversion Progress
Conversion processing is the interaction of user's source, conversion working tables and files, conversion programs, and user intervention through utility control statements.
Converted System That Can be Loaded to System/38

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 will also be flagged as requiring some user action, such as exits to System/3 Assembler subroutines.

Utilities

The System/3 copy and delete utilities are converted to System/38 control language commands by the conversion utility. All other utility functions, with the exception of the card sort utility, are supported with easier to use functions provided through the control language.

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 and RPG II and control language source must be moved to System/38 from the 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 VTOC information, the RPG file description, and input specifications. The System/3 Batch Conversion Utilities then permit the user to optionally expand and modify the data descriptions provided for him, as well as to create additional descriptions. In addition, the utility provides extensive cross-reference reports.

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

- Converted programs with control language generated for compiling them.

- Control language from the operational control language.

- Data files with data descriptions.

- Control language to create System/38 described data files and to load the data from the System/3 into the data files.

Glossary

This glossary contains terms in the System/38 publications. If you do not find the term you are looking for, see the *IBM Data Processing Glossary,* GC20-1699.

access path: The means by which the Control Program Facility 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 that specifies the maximum number of jobs that can execute concurrently in the storage pool.

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. Autostart jobs are specified for a subsystem by autostart job entries in the subsystem description.

auxiliary storage: All addressable storage space other than main storage. Auxiliary storage is located in the system's non-removable 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.

breakpoint: A place in a program (specified by a command or a condition) where the system halts execution so that the user can display variables, modify variables, or modify the execution sequence of the program. The user can use breakpoints to test his programs.

CICS: Customer information control system.

class: An object that specifies the execution parameters for a routing step. The class object is specified in the routing entry in a subsystem description.

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

command definition: An object that defines a command (including the command name, parameters, and validity checking information) and identifies the program that performs the function requested by the command.

control language: The set of all commands with which a user requests functions of the Control Program Facility.

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 that program.

controlling subsystem: The interactive subsystem that is started automatically when the system is started and through which the system operator controls the system.

control storage: High-speed storage containing microcode. This storage can be implemented as read only (ROS) or random access (RAM).

data area: A variable that is a created object and that is available to all programs (such as control language and RPG III programs) that execute using the Control Program Facility.

data authority: The right to read, modify, add, or delete data contained in an object.

data base: The collection of all the files stored in the system. Files in the data base are called data base files. See also physical file and logical file.

data description specifications: A description of data base or device files that is entered using a fixed-form syntax. The description is used to create the files.

data rights: The authority that controls how a system user can use the data contained in an object.

device description: An object that contains information describing a particular device that is attached to the system.

device file: A file that is processed on an external input or output device attached to the system, such as a work station, a card read and punch unit, a printer, or the diskette magazine drive.

EIA/CCITT: Electronics Industries Association/Consultative Committee on International Telegraphy and Telephony.

exception: The occurrence of a monitorable machine or user-defined condition directly associated with the execution of a particular function within a program. Exceptions generally represent an abnormality detected by the machine or by a program.

externally described data: Data contained in a file for which the fields in the records are described to the Control Program Facility, through the use of the data description specifications, when the file is created. The field descriptions can be used by the program when the file is processed.

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.

file: A set of related records treated as a unit and including descriptive information about the records.

file description: The information contained in the file that describes the file and its contents. The data in the file can be described to the record level (see *program described data*) or to the field level (see *externally described data*).

file overrides: Parameters that are specified when a file is used to temporarily change parameters specified when the file was created.

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

general purpose library: The library provided by the Control Program Facility to contain user-oriented, IBM-provided objects and user-created objects that are not explicitly placed in a different library when they are created.

HDX: Half duplex.

IMPL: Initial microprogram load.

IMS: Information management system.

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.

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

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 in which the attributes of a job can be predefined and stored.

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 the Control Program Facility.

job queue: A queue on which batch jobs are placed when they are submitted to the system and from which they are selected for execution by the Control Program Facility.

job stream: The sequence of one or more jobs consisting of control language commands and, optionally, data related to one or more of the jobs.

keyed sequence access path: An access path that is based on the contents of key fields contained in the records.

key field: A field, contained in every record in the file, whose contents are used to sequence the records when the file is used.

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.

library list: An ordered list of library names indicating which libraries are to be searched, and the order in which they are searched, to find an object.

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.

LU: Logical unit.

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

members: 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 work station display in which a list of the options is shown to the user. From this menu the user selects the option(s) he wishes.

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

message description: A definition of a message that provides descriptive information about the message and contains the text of the message.

message queue: A queue (associated with a person or program) on which messages are placed when they are sent to the person or program. The person or program obtains the message by receiving it from the message queue.

object: A named unit that consists of a set of attributes (that describe the object) and data. The term object is used to refer to anything that exists in and occupies space in storage 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: A list of output files that are ready to be written to an output device by a writer.

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, a physical file contains fixed-length records, all of which contain the same fields.

problem determination: The process of identifying the source of a problem as being a program component, a machine failure, data communications facilities, user or contractor-installed programs or equipment, an environment failure, or a user error.

program described data: Data contained in a file for which the fields in the records are not described through the Control Program Facility. The fields must be described in the program that processes the file.

prompt: A request for information or user action. The work station user must respond to proceed.

qualified name: The combination of an object name and a library name used to identify an object.

query: A request to extract, from a file, one or more records based upon specified criteria. For example, listing all the customers in a customer master file whose balance is greater than $1,000.00.

reader: A Control Program Facility program that reads jobs from an input device and places them on a job queue.

routing data: A character string that the Control Program Facility compares with character strings in the routing entries to select the routing entry to be initiated for 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 jobs that execute in the subsystem.

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

second-level message: Information presented to the user in addition to the first-level message. Second-level messages are obtained by pressing the Help key while a first-level message is displayed.

security: The prevention of access to or use of data or programs by unauthorized persons.

shared access path: An access path used by more than one file to provide access to data shared by the files. The access path specifications are contained in the description of each file that uses the access path.

shared data: Data that can be accessed through more than one data base file.

shared format record: A record format that is used in more than one externally described data file.

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 is not intended for direct access to a device but that provide access to data processed by the readers and writers.

storage pool: A quantity of main storage available for use by jobs executing in the storage pool. The storage pool does not consist of a given block of storage; rather it specifies an amount of storage that can be used.

subsystem: A predefined operating environment through which the Control Program Facility 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 the Control Program Facility uses to control the subsystem.

system library: The library provided by the Control Program Facility to contain system-oriented objects provided as part of the Control Program Facility.

system variable: A variable that is a created object and that is available to all programs (such as control language and RPG III programs) that execute using the control program facility.

temporary library: A library that is automatically created for each job to contain objects created by that job that are not specifically placed in another library. The objects in the temporary library are deleted when the job ends.

user identification: The ability to recognize 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 represents a particular user or group of users to the Control Program Facility. The user profile identifies which objects and functions the user is authorized to use.

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. Examples are display/keyboard devices and printer/keyboard devices.

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

Index

automatic initial microprogram load 3-6
auto report, RPG III 2-38
auxiliary storage 3-11

batch conversion utility, System/3 1-7
batch job streams 2-10

cables, work station controller
  coaxial 3-23
  twinaxial 3-22
cable thru feature 3-30
card device 3-19
cluster feature 3-31
coaxial cable 3-23
command grouping menu 2-8
commands 2-7
communication adapters 3-21
communication attachment 3-25
compatibility, with System/3 4-1
compilation listing functions, RPG III 2-37
control adapter, system 3-10
control language 2-1
control language programs 2-9
control program facility 1-3, 2-2
control program facility functions 2-3
conversion from System/3 4-1, 4-2
conversion reformat utility 1-8, 2-41
copy 2-28

data base data management 2-18
data base file processing 2-22
data base files 2-19
data description specifications 2-15
data file utility 2-40
data management
  card device 2-25
  data base 2-18
  device 2-22
  diskette device 2-25
  display device 2-23
  printer device 2-25
debug facilities, RPG III 2-38
debugging 2-26

device configuration 2-30
device control expansion 3-24
device data management 2-22
device interface expansion 3-24
disk enclosures 3-11
diskette 3-13
diskette magazine drive 3-12
display device data management 2-23
display highlighting functions, 5250 3-34
displays 2-4
display screen, system console 3-8

externally described data 2-15

file reference 2-27
files
  logical 2-19
  physical 2-19

high-level machine interface 3-3
history log 2-29

input/output devices 1-9
interactive data base utilities 1-7, 2-39
  data file 2-40
  query 2-41
  source entry 2-39
interfaces, user and machine 1-2
internal input/output channel 3-5

job log 2-29
jobs 2-12
job streams, batch 2-10

- keyboards
  - data entry, 5250 3-33
  - typewriter-like 3-32
  - system console 3-9

- language enhancements, RPG III 2-33
- libraries 2-11
- log
  - history 2-29
  - job 2-29
- logical files 2-19

- machine 1-8
- machine configurator 3-2
- machine interface, high-level 3-3
- machine summary 3-1
- magazine, diskette 3-13
- magnetic tape device 3-20
- magnetic stripe reader, 5250 3-34
- main storage capacity 3-6
- managing work 2-12
- message handling 2-28
- messages 2-30
- microprogram load, automatic initial 3-6

- operator/service panel, system console 3-10
- object management 2-10
- objects 2-10

- physical files 2-19
- power keylock 3-6
- print belts 3-15
- printers 3-15
  - 3262 3-18
  - 5211 3-17
  - 5256 3-29
- processing unit 3-4
- programmer services 2-25
- program products 1-3, 2-1

- query utility 2-41

- remote communications 3-25
- RPG III 1-6, 2-33
  - auto report 2-38
  - compilation listing functions 2-37
  - debug facilities 2-38
  - enhancements 2-33

- save/restore 2-36
- security 2-31
- selector light pen, 5250 3-34
- service 2-32
- service panel, system console 3-10
- services
  - programmer 2-25
  - system 2-30
  - system operator 2-29
- source entry utility 2-39
- spooling 2-14
- storage management 3-5
- subsystems 2-13
- system console 3-7
  - display screen 3-8
  - keyboard 3-9
  - operator/service panel 3-10
- system control adapter 3-10
- system operator services 2-29
- system services 2-30
- system unit 1-8, 3-3
- System/3 batch conversion utilities 1-7

- test environment 2-26
- testing 2-26
- time-of-day clock 3-6
- twinaxial cable 3-22

- user interfaces 2-4

- virtual address translator 3-5

work, managing 2-12
work station controllers 3-22
  basic 3-22
  coaxial cable 3-23
  device control expansion 3-24
  device interface expansion 3-24
  second 3-24
  twinaxial cable 3-22
work stations 3-26

3262 printer 3-17
3410/3411 magnetic tape subsystem 3-20
5211 printer 3-17
5250 information display system 3-26
5251 display station 3-27
5252 dual display station 3-28
5256 printer 3-29
5381 processing unit 3-4
5424 multi-function card unit 3-19

READER'S COMMENT FORM

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

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

Page Number *Error*

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

Page Number *Comment*

Note: All comments and suggestions become the property of IBM.

Name ________________________________

Address ______________________________

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

IBM System/38
Introduction

GC21-7728-0

Cut Along Line

Fold

Fold

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

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

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

Fold

Fold

IBM®

International Business Machines Corporation

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

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

IBM®

International Business Machines Corporation

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

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

System/38 Introduction (File No. S38-00) Printed in U.S.A. GC21-7728-0

GC21-7728-0