IBM

Series/1

GC34-0285-1

File No. S1-00

IBM Series/1
Programming System
Summary

Second Edition (August 1980)

This is a major revision of, and obsoletes, GC34-0285-0 and Technical Newsletter GN34-0625. This book has been enhanced to describe the new support provided by the IBM Series/1 Event Driven Executive operating system for Version 3 plus some editorial comments. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the changes.

Preface

This publication is an introduction to IBM Series/1 programming support. For an overview of Series/1 hardware support, refer to the *IBM Series/1 System Summary*, GA34-0035.

The following program categories are available for the Series/1:

- *Licensed programs*—developed by IBM to perform an end-use function for the Series/1 customer. They are distributed through the IBM Program Information Department to customers under the terms of the Agreement for IBM Licensed Programs.
- *Request for price quotation* (Programming RPQ)—developed by IBM to meet individual customer requirements. Via the programming RPQ, customers can request alterations or additions to available IBM programming support.
- *Field-developed programs (FDP)*—developed by IBM branch office personnel to perform end-use functions for the user.
- *Installed user programs (IUP)*—developed by or for an IBM system user and used to perform a variety of user functions.

This publication only describes the licensed programs and programming RPQ offerings that are available for the Series/1. For information about available FDPs and IUPs, contact your IBM representative.

Audience

The intended audience of this book is the customer executive or data processing professional responsible for evaluating the technical strengths and applicability of Series/1 programming support. To understand the Series/1 software offerings, the reader should be familiar with data processing concepts and operating systems.

Book Organization

The subject matter is presented in six chapters and two appendices:

- *Chapter 1. Series/1 Programming Support* is an overview of the Series/1 programming support offerings. It introduces the reader to the Series/1 operating systems and the software products they support. Each software product is discussed in a subsequent chapter. Data security and operating system generation and installation is also discussed.
- *Chapter 2. Series/1 Operating Systems* describes the Series/1 operating systems and program preparation support offerings:
  - Event Driven Executive
  - Realtime Programming System
- *Chapter 3. High-Level Languages and Supporting Libraries* describes the components of PL/I, COBOL, FORTRAN IV, and MFSL.
- *Chapter 4. Communication Products Support* describes the components and features of software products that support hardware communications features that allow telecommunication with other systems and devices.
- *Chapter 5. Special Device Support* describes the components of software products that support special hardware I/O features that can be attached to the Series/1 or which attach the Series/1 to the System/370.
- *Chapter 6. Data Management and Access Support* describes the components of software products that provide data file management and data access.
- *Appendix A. Control Program Support* describes the Programming RPQs used to build a tailored Series/1 operating system.
- *Appendix B. Series/1 Programming Publications Directory* is a list of manuals supporting at least all of the products discussed in this book. Each manual is identified by a full title, product program

number, and book order number. The list also identifies those manuals no longer available.

Related Publications

This publication references other publications that describe particular IBM Series/1 software products in detail. The *IBM Series/1 Graphic Bibliography*, GA34-0055, describes most of the technical publications needed by those who plan for, install, program, operate, and maintain the IBM Series/1.

This book also provides a quick reference to the title, order number, intended audience, and content of each of these publications.

For an overview of Series/1 hardware support, refer to the *IBM Series/1 System Summary*, GA34-0035

To obtain Series/1 publications, contact your IBM representative.

Contents

Chapter 1. Series/1 Programming Support

This chapter provides an overview of IBM Series/1 programming support. This support consists of three operating systems and a wide variety of other optional licensed programs that permit you to select and use only those functions required to meet your application needs.

Although the Series/1 operating systems functions overlap to some extent, each is aimed at meeting unique needs. The operating systems described in this publication are:

- *Event Driven Executive*—provides an entry level operating system with general functional coverage in a small hardware configuration. The Event Driven Executive can address multiple cooperating, or independent applications. Using the Event Driven Language (EDL), it is possible to have a Series/1 configuration that is diskette based with a 32K byte to 64K byte memory size. With the addition of high-level languages, it is possible to have a disk based Series/1 with 64K bytes to 128K bytes memory size, and beyond. A configuration with a large number of terminals requiring a variety of independent applications should be evaluated for implementation using the Realtime Programming System. Using the Event Driven Executive, a first time system user can expect to be able to develop and implement an application in a relatively short time.

  The Event Driven Executive operating system can be characterized as supporting entry level, performance oriented, Series/1 hardware configurations and capable of supporting realtime, commercial, and transaction processing applications with significant, easy to use, capabilities.

- *Realtime Programming System*—provides extensive capabilities for larger complex hardware configurations with functional depth. Due to the many options and comprehensive facilities provided, the Series/1 configuration under the Realtime Programming System can only be disk based and a 128K byte to 256K byte memory size is recommended. The emphasis of the Realtime Programming System is on capacity and multiple independent complex applications, including realtime, commercial, and transaction processing environments. Due to the dynamic and transient orientation of the Realtime Programming System, the operating characteristics of the system can adjust to the changing demands of the user environment.

System implementation of the Realtime Programming System requires a systems programmer to select and determine the level of system interface. A longer term systems development commitment and more detailed understanding is also required to take advantage of the available system function. With the introduction of the command language facility in Version 4, a higher level of user interface is available to increase application development productivity. The command language facility can be used to provide a less complex application execution environment.

- *Control Program Support*—(see Appendix A) provides the capability for an experienced systems user to select from numerous support modules and produce an efficient systems solution with the application integrated within the operating system. The higher development cost may result in smaller hardware configurations and more efficient application execution environments. Various application aids based on Control Program Support in the form of field developed programs (FDPs) and installed user programs (IUPs) are available to solve specific application requirements.

The possibility of having application and Series/1 hardware requirements that span both the Event Driven Executive and the Realtime Programming System exist. It is, therefore, the intent and capability of having both operating environments installed on separate systems within a user environment as a systems solution. The high-level languages such as PL/I, COBOL, FORTRAN IV and application tools such as Multiple Terminal Manager, Indexed Access Method, and Sort/Merge are functionally equivalent or compatible subsets within the base operating systems. An application programmer using these high-level interfaces can develop programs that can be used with either the Event Driven Executive or the Realtime Programming System.

With the Multiple Terminal Manager support available on both the Realtime Programming System and Event Driven Executive operating systems, development productivity in the area of interactive or transaction processing is greatly enhanced. The Multiple Terminal Manager provides an easy to use high-level interface for both operating systems.

Suggestions for selecting the operating system that best fits your application's needs and hardware configuration appear in the following section.

Operating System Selection

Selecting the operating system for a particular application and hardware configuration must take into account a number of different factors beyond the comparison of functional differences. Basically, the factors to be considered relate to the environment in which the operating system must work. Specifically, the following environments must be considered:

- Application
- System
- Hardware
- User

Figures 1-1 and 1-2 illustrate the characteristics of the environments which are best suited to the Realtime Programming System or Event Driven Executive operating systems respectively. The general characteristics listed in these two figures exhibit the basic differences between the two operating systems.

Realtime Programming System			
Application	**System**	**Hardware**	**User**
• Complex • Concurrent applications • Shared programs/data	• Comprehensive • Capacity • Growth • Dynamic/transient	• Complex configuration • Larger processors • Large Storage (128K to 256K bytes)	• Experienced • Systems programmer • Long development cycle

Figure 1-1. Realtime Programming System characteristics

Event Driven Executive			
Application	**System**	**Hardware**	**User**
• Entry level • Single/Cooperative applications	• General • Performance • Low overhead • Static/fixed	• Entry level configuration • Small processors • Minimum storage (32K to 128K bytes)	• First time • Minimum education • Short development cycle

Figure 1-2. Event Driven Executive characteristics

Figure 1-3 compares the functions of the Realtime Programming System and Event Driven Executive operating systems. A description of each operating system follows Figure 1-3. A more detailed description is given in Chapter 2, "Series/1 Operating Systems."

Realtime Programming System		Event Driven Executive
• Transient functions • Extensive overlay • Dynamic storage • Supervisor up to 128K • Extensive error handling • Disk resident • Shared program logic (across address spaces) • Shared "in-storage data" • Supervisor protect • Functional depth • Reentrant	Program management	• Fixed-resident • Call support • Predefined storage • Supervisor up to 64K • General error handling • Disk/diskette storage resident • Independent program logic • Common data area • Supervisor access • General coverage • Reusable
• Fixed/variable, unblocked/blocked, spanned records • System managed integrity/protection	Data management	• Fixed records (blocking optional) • Device independent within class • User managed protection
• Capacity • Manages large configurations • Depth of system function	Advantages	• Low overhead • High performance • Entry configurations

Figure 1-3. Series/1 operating system comparison summary

Event Driven Executive

The Series/1 Event Driven Executive is an operating system designed for ease of use and interactive applications. It is adaptable to a low entry multiprogramming diskette-based production operation, as well as to a disk-based multiterminal, multiuser production system.

The Event Driven Executive is available in three versions (refer to Chapter 2 for additional information about the features of each version). Each version consists of:

- A storage resident Supervisor and Emulator that provides:
  - Multitasking and multiprogramming
  - High-level instruction set
  - Device support for:
    - —Disks and diskettes
    - —Display terminals, including ASCII devices
    - —Printers
    - —Tapes
    - —Binary synchronous and asynchronous communications
    - —Sensor input and output devices
  - Other common use system functions such as:
    - —Floating-point support
    - —Timer support
    - —Printer output spooling
- A Utility product that provides a set of productivity aids, such as:
  - Data set management
  - Application program and system maintenance
  - Distributed processing and communications control
  - Source program entry and editing
  - Screen formatting
  - Job stream processor
  - Remote job entry
- A Program Preparation Facility that provides:
  - The compiler for the Event Driven Language (EDL)
  - The capability to prepare custom supervisors
- A Macro Library and a Macro Library/Host for preparing assembly language programs

Not all functions are available on all versions; newer versions contain more functions, but generally require more storage.

To meet unique customer configuration needs and to match a customer's previously developed programming skills, EDX supports a set of separately priced Series/1 licensed programs. These programs provide a selection of high-level languages and application productivity aids.

- High-level languages
  - FORTRAN IV, ANS X3.9-1966
  - COBOL Compiler and Resident Library, ANS X3.23-1974 and Low Intermediate Level, FIPS 21-1
  - PL/I, subset of ANS X3.53-1976

- Application productivity aids
  - Sort/Merge
  - Mathematical and Functional Subroutine Library
  - Indexed Access Method
  - Data Collection Interactive
  - Multiple Terminal Manager
  - System/370 Channel Attach Program

Using the System

The Event Driven Executive allows multiple, cooperating, concurrent application programs to execute on a single Series/1. Each user can write application programs with limited concern for, or knowledge of, the supervisor program. However, each user must know about the resources used by other applications. Applications supported by the Event Driven Executive are:

- Distributed processing
  - Data entry
  - Remote job entry
  - Inquiry
- Commercial applications
  - Billing
  - Inventory control
  - Accounts receivable
  - Sales analysis
- Graphic applications
- Sensor based applications
  - Data acquisition
  - Material and component testing
  - Machine and process control
  - Shopfloor control
  - Power management

System Generation and Installation

Diskettes shipped from IBM to install and generate the Event Driven Executive operating system include:

- A starter system
- Utilities
- Program preparation facility
- Supervisor object modules

Installation of the following programs provide an assembler for application modules written in Series/1 assembler instructions, and also to reassemble supervisor modules and utility programs:

- Series/1 Macro Assembler
- Macro Library (optional)

Installation of the following optional programs on a System/370 provides the ability to assemble application programs written in the Event Driven Executive language instruction set and Series/1 assembler instructions:

- Macro Library/Host
- System/370 Program Preparation Facilities for the Series/1
- System/370 Event Driven Executive Host Communications Facility

Program Numbers

Figure 1-4 lists the Event Driven Executive Version 1 and/or Version 2 supported program numbers.

Program	Version	Number
IBM Series/1 Event Driven Executive		
Basic Supervisor and Emulator	Version 1	5719-XS1
	Version 2	5719-XS2
	Version 3	5719-XS3
Utilities	Version 1	5719-UT3
	Version 2	5719-UT4
	Version 3	5719-UT5
Program Preparation Facility	Version 1	5719-XX2
	Version 2	5719-XX3
	Version 3	5719-XX4
Macro Library	Version 1	5719-LM5
	Version 2	5719-LM6
	Version 3	5719-LM7
Macro Library/Host	Version 1	5740-LM2
	Version 2	5740-LM3
	Version 3	5740-LM4
IBM Series/1 PL/I Compiler and Resident Library		5719-PL5
IBM Series/1 PL/I Transient Library		5719-PL6
IBM Series/1 FORTRAN IV Compiler and Object Support Library		5719-FO2
IBM Series/1 Mathematical and Functional Subroutine Library		5719-LM3
IBM Series/1 COBOL Compiler and Resident Library		5719-CB3
IBM Series/1 COBOL Transient Library		5719-CB4
IBM Series/1 Sort/Merge		5729-SM2
IBM Series/1 Indexed Access Method		5739-AM3
IBM Series/1 Multiple Terminal Manager	Version 1	5719-MS1
	Version 2	5719-MS2
IBM Series/1 Macro Assembler		5719-ASA
IBM Series/1 Event Driven Executive System/370 Channel Attach Program		5719-CX1
IBM Series/1 Event Driven Executive Data Collection Interactive Programming RPQ P82600		5799-TDE

Figure 1-4. Event Driven Executive and supported program numbers

Realtime Programming System

The Realtime Programming System is a flexible, full-function operating system designed for a variety of applications. The operating system controls and manages system resources—processor, storage, and devices. It is a multiprogramming, multitasking, event-driven, disk-based system that provides an environment for realtime, interactive or transaction, session, and batch applications.

The Realtime Programming System is available in four versions (refer to Chapter 2 for additional information about the features of each version). Each version provides the interface between Series/1 licensed programs, application programs, and hardware through the following:

- Operating System functions
  - Supervisor services
  - Data management services
  - Communication services
  - System and stand-alone utilities
- Program Preparation Subsystem facilities
- Command Language Facility (Version 4 only)
- High-level language facilities
  - PL/I, subset of ANS X3.53-1976
  - COBOL Compiler and Resident Library, ANS X3.23-1974 and Low Intermediate Level, FIPS 21-1
  - FORTRAN IV, ANS X3.9-1966
- Application development aids
  - Multiple Terminal Manager
  - Indexed Access Method
  - Sort/Merge

Using the System

The Realtime Programming System provides the facilities to code, assemble or compile, and execute programs online. With Version 4, through the resources of the command language facility, it supports a multiuser program preparation/execution environment. The Realtime Programming System supports 256 preemptive priority sublevels on each of four hardware levels. Among services provided by the Supervisor are: storage management, concurrent task operations (tasking), resource management and error management. Applications supported by the Realtime Programming System are:

- Distributed processing
  - Data entry
  - Remote job entry
  - Inquiry
- Commercial applications
  - Billing
  - Inventory control
  - Accounts receivable
  - Sales analysis

- Sensor based applications
  - Data acquisition
  - Material and component testing
  - Machine and process control
  - Shopfloor control
- Telecommunications
  - System/370 front end processor support
  - 4987 Programmable Communications support
  - System/370 Channel Attachment support
  - Systems Network Architecture (SNA) support

System Generation and Installation

Diskettes shipped from the Program Information Department to install and generate the Realtime Programming System operating system include:

- Starter systems (single and multiple address space)
- System generation programs and installation aids
- Utilities

The starter systems support the execution of user programs without generating a customized system. The system generation programs provide options in the following categories, to allow you to tailor the Series/1 system your application needs:

- IPL data set options
- Initialization commands
- Processor options
- Configuration options
- Service aids and error-handling options
- System performance options

Program Numbers

Figure 1-5 lists the Realtime Programming System and supported program numbers.

Program	Version	Number
IBM Series/1 Realtime Programming System	Version 1	5719-PC1
	Version 2	5719-PC2
	Version 3	5719-PC3
	Version 4	5719-PC4
Feature 2047 or 2048 (Command Language Facility)	Version 1	5719-AS1
IBM Series/1 Program Preparation Subsystem		
	Version 2	5719-AS2
	Version 3	5719-AS3
	Version 4	5719-AS4
IBM Series/1 Sort/Merge		5719-SM1
IBM Series/1 4987 Programmable Communications Subsystem Preparation Facility		5719-CS0
IBM Series/1 4987 Programmable Communications Subsystem Execution Support		5719-CS1
IBM Series/1 4987 Programmable Communications Subsystem Extended Execution Support		5719-CS2
IBM Series/1 4969 Magnetic Tape Subsystem Support		5719-TA4
IBM Series/1 Indexed Access Method		5719-AM1
IBM Series/1 5250 Information Display System Attachment Support		5719-TA1
IBM Series/1—System/370 Channel Attach Program		5719-CA1
IBM Series/1 FORTRAN IV Compiler and Object Support Library		5719-FO1
IBM Series/1 FORTRAN IV Realtime Subroutine Library	Version 1	5719-FO3
	Version 2	5719-FO4
IBM Series/1 Mathematical and Functional Subroutine Library	Version 1	5719-LM1
	Version 2	5719-LM2
IBM Series/1 PL/I Compiler and Resident Library	Version 1	5719-PL1
	Version 2	5719-PL2
IBM Series/1 PL/I Transient Library	Version 1	5719-PL3
	Version 2	5719-PL4
IBM Series/1 COBOL Compiler and Resident Library		5719-CB1
IBM Series/1 COBOL Transient Library		5719-CB2
IBM Series/1 Remote Management Utility (PRPQ)		5799-TDH
IBM Series/1 Multiple Terminal Manager (PRPQ)		5799-TCY

Figure 1-5. Realtime Programming System and supported program numbers

Control Program Support

An overview of Series/1 Control Program Support features follows. For additional information, refer to Appendix A. Series/1 Control Program Support is designed as a group of separately priced modules that can be used to build an operating system tailored to your specific requirements. For a complete list of the modules, refer to Figure 1-6.

The support offers task, storage, and data management facilities as well as I/O device support. Modules are designed for low overhead. Functional extensions to these modules allow various levels of support.

The Control Program Support modules are Programming RPQs and all use the Series/1 Base Program Preparation Facilities (5719-PA1) for program preparation (refer to Appendix A for additional information). Control Program Support (Programming RPQ 5799-TAA) is a prerequisite, and, where applicable, a 4962 Disk Storage Unit or 4964 Diskette Unit is required for error logging.

Using the System

Control Program Support supervisory functions are storage resident. The selected functional object modules (shown in Figure 1-6) are linked together to create a specific operating system environment to support application requirements. Applications supported by Control Program Support are:

- Intelligent terminal applications
- Distributed processing
  - Data entry
  - Remote job entry
  - Inquiry
- Commercial applications
  - Billing
  - Inventory control
  - Accounts receivable
  - Sales analysis
- Structured Programming Facility (see Appendix A)

System Generation and Installation

Diskettes shipped from the Program Information Department to install and generate the Control Program Support operating system include:

- Disk IPL/loader
- Disk install utility
- Error report generator utility
- Diskette IPL/loader

These programs are installed by the diskette-to-disk copy utility. The procedure to install the programs is defined, with examples, in the Program Directory letter shipped with the basic material from IBM.

Program Numbers

Figure 1-6 lists the Control Program Support modules.

Function	Program number	Name
Task and data management	5799-TAA	Control Program Support
	5799-TAL	Control Program Support Extension I
	5799-TAQ	Control Program Support Extension II
	5799-TBQ	Control Program Support Extended Function
	5799-TBT	Control Program Support Address Translator Support
	5799-TAH	Control Program Support Indexed Access Method
	5799-TBD	Control Program Support Commercial Arithmetic
Device support	5799-TAF	Control Program Support Binary Synchronous Communications
	5799-TCZ	Control Program Support 4963 Disk Support
	5799-TAE	Control Program Support 4979 Display Station
	5799-TAK	Control Program Support 4978/4979 Display Station
	5799-TDK	Control Program Support 4963/4966 Save/Restore
	5799-TAW	Control Program Support Disk Table of Contents
	5799-TAT	Control Program Support Sort/Merge
	5799-TAY	Control Program Support Disk Spooling
	5799-TBA	Control Program Support Format/Print
	5799-TBC	Control Program Support AutoCall Support
	5799-TAJ	Control Program Support 4991 Magnetic Stripe Card Reader
Services	5799-TBB	Control Program Support Operator Station/Debug Package
	5799-TBE	Control Program Support 4978/4979 Display Map

Figure 1-6. Control Program Support modules

Data Security and Integrity

The user is responsible for protecting data and programs from unauthorized or accidental modification, destruction, use, or disclosure. However, the IBM Series/1 has built-in characteristics and optional features to help you maintain adequate protection. They include:

- Parity checking of main storage data and the I/O channel data bus.
- Storage protection is provided on the 4955 Processor in both the translated and the nontranslated modes of operation. On the 4952 Processor, storage protection is provided only by the storage address relocation translator.
- Station address and terminal identification in data communication environments.
- Electronic lockout and data-protection features for display stations.
- Block checking on all data transmitted and received on binary synchronous and synchronous data-link communication channels.
- A battery backup unit to supply the power necessary to preserve the contents of processor storage during primary power interruptions.
- Use of logical addressing via the translator hardware to provide functional isolation of user programs.

Additional information about ways to ensure data security can be found in:

- *The Consideration of Data Security in a Computer Environment*, G520-2169
- *Data Security and Data Privacy Study*, G320-1370 through G320-1376

Chapter 2. Series/1 Operating Systems

This chapter discusses the Series/1 operating systems available to the Series/1 user. They are:

- Event Driven Executive
- Realtime Programming System

Event Driven Executive

The IBM Series/1 Event Driven Executive is described in the first part of this chapter. After general information about the overall operating system, specific information about the Event Driven Executive facilities and the licensed programs supported by each version is provided.

System Overview

The Series/1 Event Driven Executive is an operating system designed for ease of use in entry level application environments. It is adaptable to a low entry multiprogramming diskette-based production operation, as well as to a large, disk-based, multiterminal, multiuser, application development system.

The Event Driven Executive supports several categories of programming requirements: distributed processing, commercial, sensor based, process control, and transaction processing applications.

Basic Supervisor and Emulator

The Basic Supervisor and Emulator is the control program for the Event Driven Executive. The supervisor controls the execution of general purpose, user-written application programs. An instruction set (Event Driven Language) is also supplied for writing these programs. A key design feature is the support of multiple independent (time or event driven) applications, with minimum interaction between programs.

Storage Management, Tasks, and Programs. The supervisor provides storage management up to 256K bytes, if the Address Relocation Translator feature is installed. Without the Address Relocation Translator, the supervisor can provide storage management support for main storage sizes up to 64K bytes. Storage management is performed by defining multiple areas of real storage which are available for execution of disk or diskette resident programs. Each area is known as a *partition*. Each partition may contain multiple programs simultaneously, within the limits of the storage assigned to each partition. Partition sizes are assigned as a multiple of 2K bytes, and can be from 6K to 64K bytes.

The supervisor provides control for a number of concurrent tasks. When using the Event Driven Language, the basic unit of work for the supervisor is an instruction. Instructions are combined to form tasks. These tasks are assigned a priority by the user which is a measure of the relative importance (usually time dependent) of their function. Up to 510 unique task priorities can be used. Figure 2-1 gives an example of Event Driven

Executive multitasking.

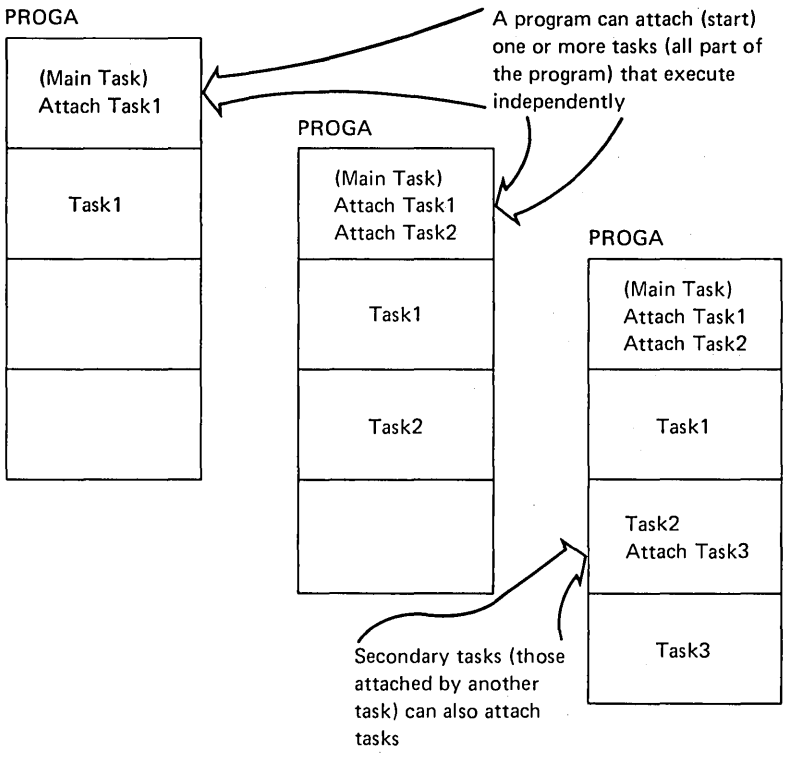


Figure 2-1. Event Driven Executive multitasking example

Task switching is performed by the supervisor whenever a higher priority task becomes available. The supervisor can be bypassed by special high data rate instructions allowing response time to be limited only by the Series/1 hardware.

Routines under supervisor control support:

- Task switching
- Device and resource queueing
- Task activation/deactivation
- Sensor I/O
- Interval timing
- Process interrupt functions

The supervisor also manages storage, communications, disk or diskette, printers, tapes, and terminals.

The multiprogramming function of the supervisor loads programs from diskette or disk into main storage. Program loading is initiated by the user from a terminal or by an executing program. Figure 2-2 illustrates the process of program loading from terminals. The program is relocated into the smallest available contiguous storage space in which it will fit, thus providing efficient storage utilization.

Programs can also be specified as overlays within a main EDL program. In this case, at main program load time, sufficient storage is reserved within the main program for the largest overlay. Overlays have two advantages

over normally loaded programs: the storage is available when needed, and fast program loading times.

For data management, routines are available within the supervisor for multiple diskettes, disk, and disks with fixed heads. File access is either sequential or direct. Multiple logical volumes can be created on any disk drive but only one logical volume per diskette is supported.

The multiterminal support provides instructions allowing interactive communications between the user and the application programs. Terminals supported are the 4978 and 4979 Display Stations, 3101 Display Terminal, 4973 and 4974 Printers, and teletypewriter equivalent terminals. Generally, the user can write terminal I/O functions within his application program without concern for the actual terminal used. The terminal to be used by the program is dynamically assigned by the supervisor as the same terminal used to initially invoke the program. Therefore, the terminal assigned can vary from one program invocation to the next without coding changes. An application program, utilizing the terminal instructions, in most cases, can be operated in a compatible manner from any terminal supported by the Event Driven Executive. A separate copy of the program will be loaded for each invocation by a terminal.

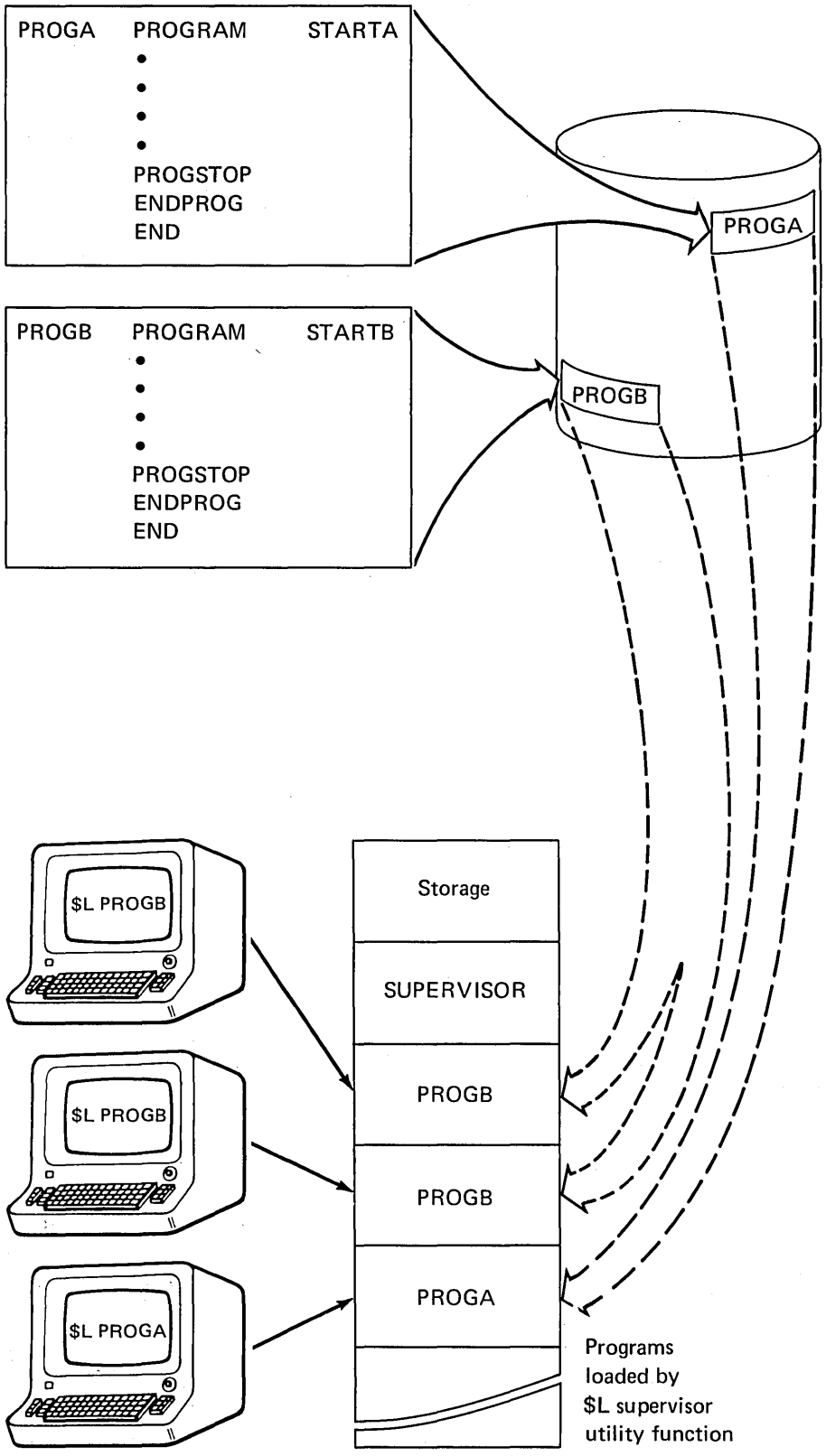


Figure 2-2. Program loading from terminals

Emulator and Instruction Set Routines. The emulator and instruction set routines within the supervisor provide support for the Event Driven Language (EDL). These instructions are designed to provide common use functions while maintaining an elemental, easily learned structure. Source programs can be written solely with these instructions and compiled using the Event Driven Executive Program Preparation Facility or the Series/1 Macro Assembler in conjunction with the native Macro Library. A text editor from the Event Driven Executive Utilities is also required, to enter the source program.

Communications

The Event Driven Executive provides support for communication applications. Communication features direct the transfer of data between programs and remote stations. A remote station can be either a terminal or another computer. The Event Driven Executive supports:

- Binary synchronous communications
- Asynchronous communications

Binary Synchronous Communications. The binary synchronous communications access method (BSCAM) provides an instruction set to send and receive data on a BSC line. Its features include:

- Multiple line support
- Point-to-point leased line
- Point-to-point switched line
- Multipoint master/tributary stations
- Optional transparent/conversational modes

BSC permits connection between:

- Series/1 to Series/1
- Series/1 to System/360, System/370

BSC permits a user-written protocol for:

- Series/1 to IBM 5100/5110
- Series/1 to System/3, System/32, System/34

Asynchronous Communications. The asynchronous communications feature is implemented through the standard terminal interface. Its support can be used for:

- Direct connect terminals
- Leased lines using modems
- Switched lines using modems

Sensor Input/Output Support

The Event Driven Executive supports the Series/1 sensor I/O devices. The following functions are available:

- Analog input/output, digital input/output, process interrupt
- Sequential and random addressing of devices
- External synchronization
- Sharing device groups and subgroups between programs

- Relay or solid state multiplexing
- Multi-range analog input
- A program for testing your sensor I/O devices

Utilities

The Event Driven Executive Utilities is a set of programs providing productivity aids for Series/1 program development, program maintenance, and distributed processing functions to a host System/360 or 370. These programs are independent program load modules capable of running concurrently with other application programs on a Series/1 and with other utility programs. They include:

Session Manager. The session manager provides menus to aid in using the facilities of the system utilities and/or user programs. Independent default environments for independent users are also provided.

Data Management. The data management programs provide functions to define, patch, dump, delete, rename, compress, copy, list, and initialize data sets on any logical volume.

Distributed Processing. The distributed processing programs provide a fully supported distributed system capability in conjunction with a host System/360 or 370. Communications to the host are via a point-to-point leased line and the binary synchronous communication single line control in the Series/1. Programs can be transferred bidirectionally between the host direct access data sets and Series/1 main storage or direct access data sets.

A remote job entry capability allows the user, through a Series/1 terminal, to invoke batch program execution on the host system. Additional distributed processing capabilities are provided via a standard RJE workstation utility emulating a 2780 or 3780. Also available is the remote management utility, which provides functions such as:

- File allocation and transfer
- Remote system interaction between the remote management utility and an application program that executes on the host system

Source Program Entry and Editing. The source program entry and editing utility provides a functional and syntactical subset of the System/360 and 370 OS/TSO text editing facility in either line, or full screen editing modes. The full screen editor is a subset of the Structured Programming Facility (SPF) of TSO. The full screen editor provides browse, edit, and merge functions. In addition, commands are included for interprocessor communications so that host program development and assembly can be controlled from a Series/1 terminal. A line editor is also provided.

Interactive Program Debugging. The interactive program debugging program, running on a Series/1, provides a tool to debug and test application programs. Program debugging is available to and interactive with any input/output terminal and can be invoked and run concurrently with other applications. The most important features of this utility are its capabilities to selectively stop, modify, trace, and restart an application

program with little or no impact on other programs currently executing on the Series/1.

Program Library Update. The program library update utility is used to add new program load modules to the program libraries.

Sensor I/O Test. The sensor I/O test functions program is used to exercise the Sensor I/O (AI, AO, DI, DO, PI) devices on a Series/1 and to list actual hardware features installed on the Series/1. The sensor I/O test program performs functions such as read/write digital input/output, write digital output with selected time intervals, and read/write analog input/output. During any exercising function, which can be selected via a terminal command, trace printing is done at the terminal.

Graphics Display Processor. The graphics display processor is a set of programs that provide the capability of generating, maintaining, displaying, and storing fixed graphic backgrounds in files. Access to these background files is available to application programs. Realtime data can be superimposed by the application program over the displayed fixed graphic backgrounds.

Screen Format Builder. The screen format builder utility can be used to design formatted screen images for the 4978 and 4979 Display Stations and the 3101 Display Terminal. The images are saved in disk or diskette data sets for later retrieval by application programs.

Job Stream Processor. The job stream processor utility provides a batch job processing facility which can be invoked concurrently with other programs. The job stream processor allows a series of programs to execute, without operator intervention.

Remote Job Entry. The remote job entry utilities simulate either a 2780 or 3780 workstation. The utility is controlled by a set of attention commands to send a job stream to the host, define disk or diskette or printer output devices, and/or spool output to disk. A companion utility prints spooled output produced by the RJE utility.

Binary Synchronous Communications Trace. The binary synchronous communications trace utility provides a means of tracing the I/O activity on a given BSC line. All trace data is stored in a disk or diskette data set and can be printed with a companion utility. Multiple lines can be traced concurrently by loading multiple copies of the trace utility.

Program Preparation Facility

The Event Driven Executive Program Preparation Facility is a set of programs that offers support for assembling Event Driven Language (EDL) source programs.

The Program Preparation Facility consists of the following programs:

- Instruction set assembler
- Assembler listing program

These programs allow you to compile application programs concurrently with the execution of other programs. You can also reconfigure and compile custom supervisors online. As long as you code only in Event Driven Language, all application development can be performed online without a macro library. If Series/1 assembler instructions are included in the program, it will be necessary to use the Event Driven Executive Macro Assembler and the native Macro Library. As an alternative, the System/370 Host Preparation Facility and Host Macro Library can be used if a System/370 is available.

The features of the Program Preparation Facility are:

- Program Preparation Facility programs can run concurrently with other programs.
- Multiple copies of the assembler and listing program can run concurrently.
- Programs can load into any available storage.
- Any terminal can be used to invoke the program.
- All references to programs and files are by symbolic names.

Although any of the Program Preparation Facility programs can execute from a diskette-based system, the limitations of file space restricts the program preparation capability. A disk-based system is recommended for an efficient, full capability development system.

Assembler. The Program Preparation Facility assembler can only assemble programs written in the Event Driven Language instruction set. It provides an online program preparation capability that can be executed concurrently with other programs. The assembler can operate on a disk or diskette based system. With this assembler you can:

- Compile or link-edit concurrent with other program execution
- Build custom supervisors online
- Perform multiple concurrent compilations
- Support online development
- Use symbolic addressing for hardware devices and data files

Linkage-Editor. The linkage-editor provides the capability of combining two or more input object modules into a single output object module.

- *Input*. Linkage-editor input consists of a data set containing control records, the various data sets containing the object modules to be linked, and an optional autocall data set.
- *Output*. Linkage-editor output consists of a printed listing and an output object module. The printed listing consists of:
  - A start message
  - Input control records
  - Any unresolved EXTRN or WXTRN labels
  - A length/ending message
  - A map of program sections and entry point locations in the output module, if required

Figure 2-3 illustrates the program preparation process of assembling or compiling, link editing, and updating. The programs can be invoked individually by the session manager or through the batch job stream processing utility.

The linkage editor is part of the Version 1 or Version 2 Program Preparation Facility; in Version 3, the Program Preparation Facility is part of the Utilities.

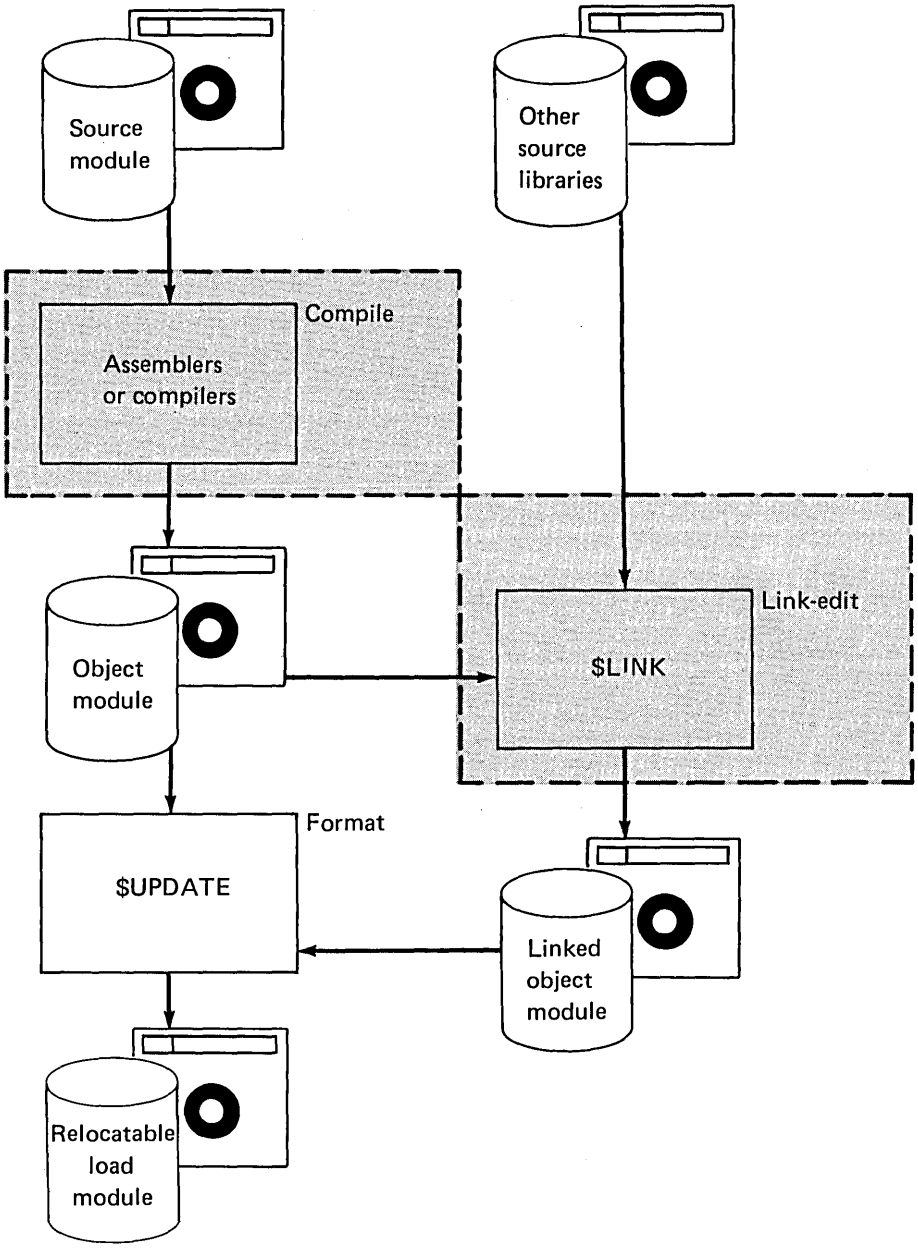


Figure 2-3. Event Driven Executive Program preparation

Macro Assembler Facility

The Event Driven Executive Macro Assembler facility provides an assembler for Series/1 instructions and/or macros. These programs can be executed online concurrent with any other application program. When used in conjunction with the native Macro Library, Event Driven Language statements can be assembled just as is done with the Event Driven Executive Program Preparation Facility.

Macro Library. The Event Driven Executive Macro Library is a set of libraries and procedures that can be used to build a Basic Supervisor and Emulator and to assemble application programs with the Event Driven Executive Macro Assembler.

Host Program Preparation Facility

Event Driven Executive programs can also be assembled by the System/370 Program Preparation Facility.

The System/370 Program Preparation Facility provides a macro assembler which permits the assembly of both Event Driven Language instructions and Series/1 assembler language instructions. This assembler requires the installation of the Macro Library/Host.

Macro Library/Host. The Event Driven Executive Macro Library/Host is a set of libraries and procedures required to assemble, on a host System/370, application programs written in Event Driven Language instructions and/or Series/1 assembler instructions.

Event Driven Language

The Event Driven Language (EDL) is a high level, easy to use programming language that is used to write general purpose programs to execute under the Event Driven Executive operating system. The Event Driven Language provides such functions as:

- High-level, easy-to-learn instructions
- Structured programming facilities
- User exits to Series/1 assembly language
- Symbolic hardware device and data file addressing
- Instructions for interactive communications
- Comprehensive instruction set
- Floating point calculation support

Program Structure. An Event Driven Language program begins with a PROGRAM statement and ends with an ENDPROG and END statement. The basic unit of a program is a task. The PROGRAM statement defines a task, which is referred to as the *initial task* of the program. Many tasks may be active concurrently and asynchronously in a program. A task can be activated or attached from the initial task or from other tasks. Any combination of instructions can be used within a task which will be executed independently from other tasks.

Tasks within a program can communicate with each other through common storage areas or through system instructions and event control blocks. The

facilities of the Event Driven Executive supervisor provide the capability of synchronizing task execution.

Instruction Set. The Event Driven Language instructions are divided into the following categories, according to their general use:

- Task control
- Data manipulation
- Program control
- Data definition
- Data formatting
- Program sequencing
- Timing functions
- Sensor based I/O
- Queue Processing
- Tape I/O
- Disk I/O
- Terminal I/O
- Graphic functions
- EXIO control
- Program module sectioning
- Listing control

Event Driven Executive Version 1 Features

Event Driven Executive Version 1 features include:

- Address Translator
  - Support for 64K-byte user partitions
  - Support for the 4952 Processor with native translator up to 128K bytes and the 4955 Processor with the relocation translator feature up to 256K bytes
  - An additional instruction to move data across partition boundaries
  - Optional mapping of some or all of the Supervisor's common areas in each partition
- Online utilities
- High-level language and instruction set
- Communications support with binary synchronous communications
- Program preparation facilities with host program preparation facility options
- Device support for:
  - 4952 Processor
  - 4953 Processor
  - 4955 Processor
  - 4962 Disk Storage Unit
  - 4963 Disk Subsystem
  - 4964 Diskette Unit
  - 4966 Diskette Magazine
  - 4973 Line Printer
  - 4974 Matrix Printer
  - 4978 or 4979 Display Station
  - 3101 Display Terminal
  - 2741 Communications Terminal
  - Teletypewriter-like devices
  - 4980 Sensor I/O Unit
- Support for the 4952 clock/comparator
- Support for the following Series/1 hardware features (multiples supported)
  - 1560 Integrated Digital Input/Output
  - 1610 Asynchronous Communications Single-Line Control
  - 2074 Binary Synchronous Communications Single-Line Control to 9600 bps
  - 2075 Binary Synchronous Communications Single-Line Control—High Speed to 56K bps
  - 2091 Asynchronous Communications 8-Line Control
  - 2092 Asynchronous Communications 4-Line Adapter
  - 2093 Binary Synchronous Communications 8-Line Control
  - 2094 Binary Synchronous Communications 4-Line Adapter
  - 2095 Feature—Programmable 8-Line Control
  - 2096 Feature—Programmable 4-Line Adapter
  - 3920 Floating Point (4955 Processor only)
  - 7840 Timers
  - 7850 Teletypewriter Adapter

Supported Licensed Programs

The Event Driven Executive Version 1 supports the following licensed programs:

- IBM Series/1 Event Driven Executive PL/I Compiler and Resident Library (5719-PL5)
- IBM Series/1 Event Driven Executive PL/I Transient Library (5719-PL6)
- IBM Series/1 Event Driven Executive COBOL Compiler and Resident Library (5719-CB3)
- IBM Series/1 Event Driven Executive COBOL Transient Library (5719-CB4)
- IBM Series/1 Event Driven Executive FORTRAN IV Compiler and Object Support Library (5719-FO2)
- IBM Series/1 Event Driven Executive Sort/Merge (5719-SM2)
- IBM Series/1 Event Driven Executive Indexed Access Method (5719-AM3)
- IBM Series/1 Event Driven Executive Mathematical and Functional Subroutine Library (5719-LM3)
- IBM Series/1 Event Driven Executive Multiple Terminal Manager (5719-MS1)
- IBM Series/1 Event Driven Executive Data Collection Interactive Programming RPQ P82600 (5799-TDE)

Event Driven Executive Version 2 Features

Event Driven Executive Version 2 provides all the facilities of Version 1 plus:

- The 4969 Magnetic Tape Subsystem is supported at the Read/Write and EXIO level of interface. Event Driven Executive FORTRAN IV, COBOL, and PL/I provide support at the Read/Write level. The Event Driven Language (EDL) provides both the Read/Write and EXIO levels of access. The Series/1 macro assembler language applications can use both Read/Write and EXIO support through EDL. Sort/Merge supports tape for SORTIN and SORTOUT.
- Remote management utility—the remote management utility provides the capability to control a remote Series/1 from a host system. The utility operates as a task under the Event Driven Executive on the Series/1 and provides an interface to the host system connected through binary synchronous communications.

Supported Licensed Programs

The Event Driven Executive Version 2 supports the following IBM Series/1 licensed programs:

- IBM Series/1 Event Driven Executive PL/I Compiler and Resident Library (5719-PL5)
- IBM Series/1 Event Driven Executive PL/I Transient Library (5719-PL6)
- IBM Series/1 Event Driven Executive COBOL Compiler and Resident Library (5719-CB3)
- IBM Series/1 Event Driven Executive COBOL Transient Library (5719-CB4)
- IBM Series/1 Event Driven Executive FORTRAN IV Compiler and Object Support Library (5719-FO2)
- IBM Series/1 Event Driven Executive Sort/Merge (5719-SM2)
- IBM Series/1 Event Driven Executive Indexed Access Method (5719-AM3)
- IBM Series/1 Event Driven Executive Mathematical and Functional Subroutine Library (5719-LM3)
- IBM Series/1 Event Driven Executive Multiple Terminal Manager (5719-MS1)
- IBM Series/1 Event Driven Executive Data Collection Interactive Programming RPQ P82600 (5799-TDE)

Refer to Chapters 3 through 6 for additional information about the features of these programs.

Event Driven Executive Version 3 Features

The Event Driven Executive Version 3 provides all the facilities of Version 2, plus support for:

- Printer output spooling. The output spooler provides management of printers, permitting more than one program to appear to be using the same printer at the same time. The spooler places data to be printed on the disk and prints it when the printer is available.

  Spooling also supports functions such as multiple copies, pausing for forms changes, holding all or part of the spool contents for subsequent printing, and restarting the printed output. Through these facilities, spooling provides more efficient management of the printers and minimizes the pre-execution planning for applications that will execute concurrently.

- Data Management Extensions. In Versions 1 and 2, data sets and volumes were limited to less than 8 million bytes. Version 3 provides support for larger data sets and volumes. The upper limit is either 2+ billion 256-byte records or the capacity of the direct access device upon which the data set resides.

  In Versions 1 and 2, the table of contents for each disk was storage resident and could only be modified through the system generation process. In Version 3, the table of contents is moved from storage to each fixed disk device. You define the volumes on a disk via a utility, instead of via the system generation process. Existing utilities are used to divide these volumes into data sets.

  In addition, diskette capacity is increased approximately 15% by providing support for 256-byte formatted sectors.

  These enhancements will provide you with significant flexibility, capacity, and ease-of-use extensions.

- Additional 3101 Display Terminal support. The IBM 3101 Display Terminal (Models 20, 22, and 23, in block mode) is supported by the terminal interfaces of EDX Version 3. Existing applications that use only roll screens (line-by-line) will, in most cases, be able to use a 3101 (Models 20, 22, or 23, in block mode) with no coding changes. Existing applications that use the full static-screen support for the 4978/4979 Display Stations, provided by the $IMAGE utility and the static screen subroutines of EDX, can use the 3101 display, in block mode, with minor coding changes.

  Since the 3101 display (Models 20, 22, and 23) can be attached through asynchronous communications interfaces, remote display terminals can be an integral part of your application solution. The session manager facilities are available for use with your 3101 display terminals. The 3101 Model 23 has an RS422A capability that is not supported.

  The 3101 (Models 10, 12, and 13) support is expanded to include support for the eight function keys—seven (PF1–PF7) for use by applications and one (PF8) used as the attention key. These function keys are also supported for Models 20, 22, and 23, in either character or block mode.

- Series/1 to Series/1. The Series/1 to Series/1 Attachment (RPQs D02241 and D02242) allows an application to be separated into two

or more processors in a single cluster. Each processor can then be dedicated to a unique operation, e.g., one handling data collection, another supplying computational support, and a third could be managing printed output, communication lines, or background type operations. This provides additional capacity and availability that has not been previously available on EDX.

- General Purpose Interface Bus (GPIB). The General Purpose Interface Bus (RPQ D02118) is a cycle stealing adapter that supports the IEEE Standard 488-1975. This interface facilitates the connection of instruments to the Series/1. These instruments could be OEM printers, plotters, graphics display units, card readers, and programmable laboratory equipment (such as digital voltmeters, signal generators, and frequency analyzers). Thus, with Version 3, additional applications are possible in manufacturing, research, process control, and medical laboratories.

Supported Licensed Program

The Event Driven Executive Version 3 supports the following IBM Series/1 licensed programs:

- IBM Series/1 Event Driven Executive PL/I Compiler and Resident Library (5719-PL5)
- IBM Series/1 Event Driven Executive PL/I Transient Library (5719-PL6)
- IBM Series/1 Event Driven Executive COBOL Compiler and Resident Library (5719-CB3)
- IBM Series/1 Event Driven Executive COBOL Transient Library (5719-CB4)
- IBM Series/1 Event Driven Executive FORTRAN IV Compiler and Object Support Library (5719-FO2)
- IBM Series/1 Event Driven Executive Sort/Merge (5719-SM2)
- IBM Series/1 Event Driven Executive Indexed Access Method (5719-AM3)
- IBM Series/1 Event Driven Executive Mathematical and Functional Subroutine Library (5719-LM3)
- IBM Series/1 Event Driven Executive Multiple Terminal Manager (5719-MS2)
- IBM Series/1 Event Driven Executive Data Collection Interactive Programming RPQ P82600 (5799-TDE)
- IBM Series/1 Event Driven Executive System/370 Channel Attach Program (5719-CX1)

Refer to Chapters 3 through 6 for additional information about the features of these programs.

System Requirements

The minimum system requirements to support system generation are:

- Processor—IBM 4952, 4953, or 4955 Processor with at least 48K bytes of processor storage (Version 3 requires 64K bytes of storage)
- Operator Station—An IBM 4978 or 4979 Display Station, IBM 3101 Display Terminal, or teletypewriter device

Note: If the teletypewriter is used, the processor must be equipped with the Teletypewriter Adapter Feature (#7850).

- Disk—either a 4962 or 4963 (all models)
- Diskette—either a 4964 or 4966 or the diskette included with the 4962 Model 2 or 2F (combination disk/diskette unit)
- Printer—either the IBM 4973 Line Printer or the IBM 4974 Matrix Printer

The optional hardware (not all versions support all hardware) that is available for use with the Event Driven Executive is:

- 4969 Magnetic Tape Subsystem (Version 2 only).
- Hardware Floating Point (Feature code 3920) support on 4955 only. This is a requirement for floating-point arithmetic support with the Event Driven Language, FORTRAN IV, or PL/I.
- Timer support (#7840). The 4952 native timer is also supported.
- Integrated Digital I/O Nonisolated (#1560).
- Multiple Sensor I/O support for the 4982 Sensor I/O Unit:
  - 1060 Analog Input Control
  - 1070 Amplifier Multirange
  - 4949 Multiplexer/Reed-Relay
  - 4950 Multiplexer/Solid-State
  - 1065 Analog Output
  - 3525 Digital Input/Process Interrupt Nonisolated
  - 3532 Digital Input/Process Interrupt Isolated
  - 3535 Digital Output Nonisolated
- Communications features:
  - Asynchronous communications:
    - —1610 Asynchronous Communications Single-Line Control (9600 bps)
    - —2091 Asynchronous Communications 8-line Control
    - —2092 Asynchronous Communications 4-line adapter
    - —2095 Feature—Programmable 8-line Communications Control
    - —2096 Feature—Programmable 4-line Communications Adapter
  - Binary synchronous communications:
    - —2074 Binary Synchronous Communications Single-Line Control (9600 bps)
    - —2075 Binary Synchronous Communications Single-Line Control/High Speed (56,000 bps)
    - —2093 Binary Synchronous Communications 8-Line Control
    - —2094 Binary Synchronous Communications 4-Line Adapter
- Series/1 to Series/1 Attachment, RPQ D02241 and D02242.
- General Purpose Interface Bus, RPQ D02118.

Realtime Programming System

The following introduces the IBM Series/1 Realtime Programming System Versions 1 through 4. The first part of this section provides general information about the operating system. The next four sections provide specific information about the Realtime Programming System features and Series/1 licensed programs supported with each version.

System Overview

The Realtime Programming System is a full-function operating system upon which applications are built and executed. It is flexible and is suitable for a wide variety of large complex applications. The operating system controls and manages system resources including–processor, storage, and devices. It is a multiprogramming, multitasking, event-driven, disk-based system that provides a multiterminal and multiuser environment for realtime, interactive, transaction, session, and batch applications.

The following introduces the features and functions of each of the Realtime Programming System components are introduced in this section:

- Supervisor services
- Data management services
- Communications
- Utilities

Supervisor Services

The supervisor controls the allocation and distribution of the physical resources of the system: storage, processor and devices. It manages programs and their interaction through a set of system management services that manipulate resources and programs on a logical level.

Storage, Task Sets, Tasks, and Programs. *Physical storage* is the total memory available in the Series/1 processor. *Logical storage*, a conceptual storage layout, provides 16-bit addressing support for up to 256K of physical storage (depending on the processor and operating system version used). Storage requirements are specified in terms of logical storage. The actual physical storage is assigned via the *Storage Address Relocation Translator* feature, which converts logical storage addresses into physical storage addresses.

Logical storage is divided in up to eight address spaces, with logical addresses of up to 64K bytes each (IBM 4955 Processor only). The IBM 4952 Processor has only two address spaces. Multiple partitions can be defined in each address space (except in address space(s) containing the system partition). A *partition* is a portion of logical storage reserved for the execution of one task set at a time. Partitions begin and end on 2K boundaries. There may be only 1 partition (the system partition) or as many as 16 (the system partition and 15 user partitions). In the multiple address space management environment, the *system partition* is reserved for the system task set and must reside in address space 0; it also resides in address space 1 for an I/D (instruction/data) split system. The single address space management environment has supervisor and user partitions in address space 0.

Primary storage is the addressable physical storage associated with a partition, while *secondary storage* is the unaddressable physical storage (beyond 64K) associated with a partition. Secondary storage is used for storage overlays and is optional for a partition.

Applications reside in partitions as task sets, and only one task set can occupy a partition at a time. Before a task set can be loaded into a partition for execution, it must reside in a data set on a disk or diskette volume called a *task set library*. An application program can use one or more I/O devices or data sets that are described symbolically within the program by DSDs (refer to the data management section for additional information about DSDs). For improved performance of application programs, the operating system can *prebind* a task set to a partition and to physical devices and data sets before the task set is activated.

A task set is composed of one or more tasks. Each task is composed of one or more programs. A program can be a disk resident transient, a disk resident overlay, a storage overlay (resident in the secondary storage defined for that partition), or simply storage resident (resident in the primary storage defined for that partition). Task sets are initiated under program or operator control.

The *task* is logically the basic execution unit under the operating system, and is limited in number only by the storage available. Tasks are started by a request from another task. Synchronization and communication between tasks is made possible by the services provided for managing events, queues, serially reusable resources, timers, and interrupts. Tasks are the basis for allocation of the processor and other system resources. Tasks compete for all resources based on a priority assigned by the user.

Figure 2-4 shows the relationship of a task set, tasks, and programs. For more detailed information about the supervisor services, refer to the *Supervisor Macro Programmer's Guide* for your version of the system.

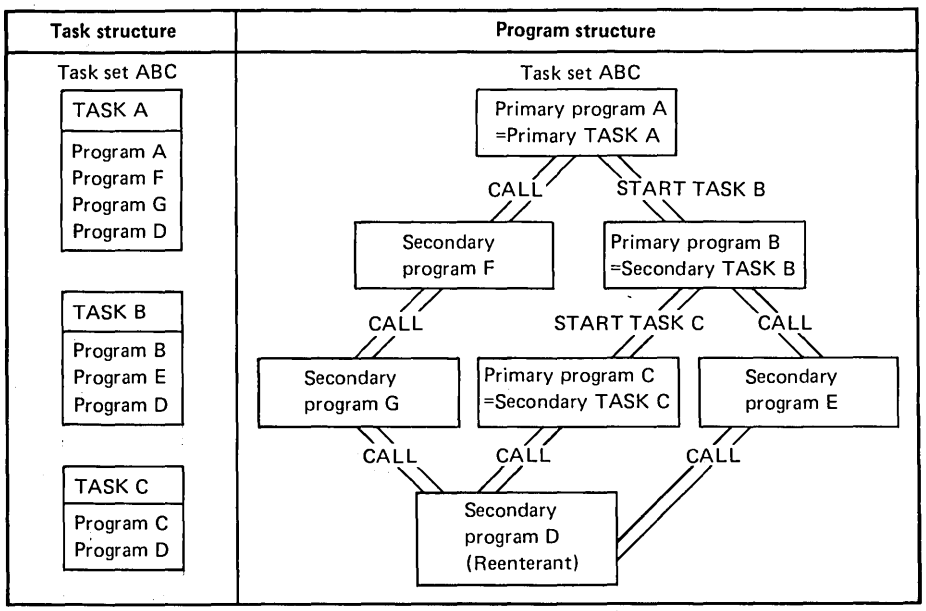


Figure 2-4. Relationship of Task Set, Tasks, and Programs

Data Management

Data management is a part of the operating system that moves information between processor storage and external devices, maintains the data on those devices, and controls those devices. Data management is divided into two categories: *data set management* and *device management.*

Data Set Management. Data set management maintains data sets and provides access to all data sets on all devices. All access to data sets or devices is through the data set definition (DSD), which describes the data set or device and how a program can gain access to it. A DSD can be in the using program or, optionally, in a DSD table (DSDT) data set in the task set library. The DSDT provides a method to maintain device independence. A DSD in the DSDT can be changed through utility commands, the Command Language Facility, or job stream processor control statements. This allows you to change the device or data set attributes without changing any code in the program. Data set management provides:

- Three data set organizations—there are three types of data set organization—consecutive, random, and partitioned. These organizations are available at the physical and the logical access levels. Random and partitioned organizations are available on direct-access devices only. Consecutive organization is available for any device. The organization of a disk or diskette data set is determined when you create the data set.
- Three levels of access—there are three levels of access to a data set—basic, physical, and logical. The level chosen is determined by the macro that will be used to gain access to the data set.
  - Basic (EXIO)—access to data on a device by physical record
  - Physical (READ/WRITE)—access to data on a device by block
  - Logical (GET/PUT)—access to data by logical record with automatic blocking and buffer management
- Three access methods—access methods are techniques for moving data between processor storage and I/O devices. Data set management has two access methods—sequential and direct (the Indexed Access Method is also supported through a licensed program) The access method is determined when the data set is opened. Both access methods can be used at the physical or logical access level. Any I/O device can use sequential access, but only a display station or a direct-access device, such as a disk or diskette, can use direct access. You choose the access method to be used depending on the application of the data set.

Note: Indexed Access Method support provides access to records in a data set by a predefined field called a key. (For further information about the Indexed Access Method support, refer to Chapter 6.)

Device Management. Device management controls the actual transfer of data to or from a I/O device. Using a set of device handlers which are part of the system task set. Device handlers are started when the system is loaded through 'STDV' commands which are placed in an IPL data set, or when the request to start a device is issued.

The types of devices that are supported by device management are:

- Data-Processing I/O. The data-processing I/O portion of device management controls the following devices through either a READ/WRITE, GET/PUT, or EXIO interface:
  - Operator station (display station or teletypewriter)
  - Other display stations
  - Matrix printer
  - Line printer
  - Diskette unit
  - Fixed-disk storage unit
- Automatic Device Backup. Automatic device backup is available for the matrix and line printers and the teletypewriter. If automatic device backup is selected and an unrecoverable error occurs on one of those devices, the operating system will automatically switch all output requests to the alternate device. (For more information on specifying automatic device backup, refer to the *Generation and Installation Procedures* manual.)
- Sensor I/O. The sensor I/O portion of device management controls the following devices through either a READ/WRITE or an EXIO interface:
  - Digital input
  - Digital output
  - Analog input
  - Analog output
- Timer I/O. The timer I/O portion of device management supports:
  - Time-of-day dependent operations
  - Time interval operations; either single or repetitive

Access to timers is through either a READ/WRITE or an EXIO interface.

For a more detailed information about data management, refer to the *Data Management Macro Programmer's Guide.*

Communications

The Realtime Programming System has routines to support applications that use communications. The purpose of a communications system is to make the power of a computer available to users working in remote locations. To achieve this, a communications system must do two things:

- Transmit data between the computer and the remote location
- Process data in the computer

The term Series/1 Communications System is used in this manual to specify the part of a Series/1 system actively involved in transmitting data between the Series/1 computer and the remote location. The Series/1 communications system is made up of:

- Series/1 computer
- Remote stations
- Communications lines
- Communications user programs
- Series/1 Realtime Programming System communications support

Note: When a Series/1 is communicating with another Series/1, the same version is not required on different computers. For remote IPL, however, both the host Series/1 and the remote Series/1 must have been upgraded to at least Version 2 of the Realtime Programming System.

The Series/1 computer also includes the line adapters connecting the processor to the communications lines. A remote station can be a terminal or another computer. For example, the Series/1 Realtime Programming System Version 4 supports communications with:

- System/370 BTAM DOS/VS
- System/370 BTAM DOS/VS CICS/VS
- System/370 BTAM OS/VS1
- System/370 BTAM OS/VS1 CICS/VS
- System/370 BTAM OS/VS1 IMS/VS (via IRSS)
- System/370 BTAM OS/VS2 MVS
- System/370 BTAM OS/VS2 SVS
- System/370 TCAM OS/VS1 (except conversational mode)
- System/370 TCAM OS/VS2 (except conversational mode)
- System/370 VTAM DOS/VS
- 2740 Communications Terminal (Models 1 and 2)
- 2741 Communications Terminal
- 3271 Control Unit (Models 1 and 2)
- 3275 Display Station (Models 1 and 2) with or without dial feature
- 3277 Display Station (Models 1 and 2)
- 3284 Printer (Models 1, 2, and 3)
- 3286 Printer (Models 1 and 2)
- 3288 Printer (Model 2)
- ASR 33/35 Teletype (trademark of the Teletype Corporation)
- System/3 CCP/RPG

- System/3 RPG
- Another Series/1
- System/32
- System/34 RPG and BSCEL
- 5260 Retail Terminal System (OS/VS1 - OS/VS2 TCAM, OS/VS1, OS/VS2, DOS/VS CICS)

Utilities

The Series/1 utilities are part of the operating system. They are IBM-supplied programs with which you can easily and efficiently manage data and maintain your system. Realtime Programming System utilities can be divided into two categories—*stand-alone* and *system* utilities.

The following lists the utilities that are available:

Stand-Alone Utilities. The stand-alone utilities are loaded from diskette, and no other program can execute concurrently with them.

The processor storage-to-diskette dump utility resides by itself on a diskette, and runs on any Series/1 with a diskette unit and at least 16K bytes of storage. The disk initialization, UTS update, and system build utilities are supplied on a diskette along with a stand-alone utility monitor.

The stand-alone utilities are:

- Processor storage-to-diskette dump
- Disk initialization
- System build
- UTS update
- Save disk to diskette
- Restore diskette to disk

System Utilities. The system utilities run under the operating system. When invoked, the system utilities are loaded as a user task set into a partition.

System utility commands can be executed using an interactive device (such as an operator station) or a noninteractive device (such as a disk data set). The system utilities are:

- COMPRESS
- COPY
- DEFINE
- INITIALIZE
- IPLMAINT
- MERGE
- PATCH
- REPORT

Realtime Programming System Features

The Realtime Programming System Version 4 is a culmination of all the features of Versions 1 through 3, plus significant productivity enhancements (notably the command language facility), and greater functional support. In order to present the detail progression of function, the following pages describe the Versions as a progression.

The Realtime Programming System Version 1 (5719-PC1) features include:

- Supports up to 64K bytes of physical storage.
- Supervisor services for the management and control of all resources.
- Data management directs the transfer of data between programs and I/O devices—either sensor-based I/O or data-processing devices. The operating system has three levels of access for application programs. They are:
  - Basic (EXIO)
  - Physical (READ/WRITE)
  - Logical (GET/PUT)

  Two access methods:
  - Sequential
  - Direct

  Three types of data set organizations are also supported:
  - Consecutive
  - Random
  - Partitioned
- Communications support for directing the transfer of data between your programs and remote stations. A remote station can be either a terminal or another computer. Communications is the part of the operating system that:
  - Handles point-to-point connections between stations that use start-stop and binary synchronous communications line control.
  - Establishes, terminates, and controls access between your programs and remote stations.
  - Transfers data between your programs and remote stations on point-to-point lines.

  The stations used by start-stop communications are (1) the 2740 Communications Terminal Model 1 (switched and nonswitched), and (2) the Teletype Model 33/35, or equivalent (nonswitched). Communications between a Series/1 and a System/370 using OS/VS1 or OS/VS2 (SVS or MVS) BTAM is supported through BSC line control in point-to-point connections.
- Utilities support for installation and maintenance of application programs and data. There are stand-alone utilities that are loaded from diskette and system utilities that reside on disk. The system utilities can run concurrently with applications; application programs cannot run concurrently with stand-alone utilities.

Compatibility

The Realtime Programming System Version 1 is upward compatible with Version 2. Problem-state programs that compile and execute on the Realtime Programming System Version 1 will also compile and execute on Version 2.

The Program Preparation Subsystem Version 1 is required to prepare output programs to execute on the Realtime Programming System Version 1.

Supported Licensed Programs

The Realtime IBM Series/1 Programming System Version 1 supports the following IBM Series/1 licensed programs:

- IBM Series/1 Program Preparation Subsystem Version 1 (5719-AS1)
- IBM Series/1 Realtime Programming System PL/I Compiler and Resident Library Version 1 (5719-PL1)
- IBM Series/1 Realtime Programming System PL/I Transient Library Version 1 (5719-PL3)
- IBM Series/1 Realtime Programming System FORTRAN IV Compiler and Object Support Library (5719-FO1)
- IBM Series/1 FORTRAN IV Realtime Subroutine Library Versions 1 (5719-FO3) and 2 (5719-FO4)
- IBM Series/1 Realtime Programming System Mathematical and Functional Subroutine Library Version 1 (5719-LM1) and 2 (5719-LM2)

Additional programming support is provided through the following Programming RPQs:

- IBM Series/1 Realtime Programming System Basic Sort
- IBM Series/1 Realtime Programming System Remote Job Entry
- IBM Series/1 Realtime Programming System 4978 Display Station Support
- IBM Series/1 Realtime Programming System Disk Spooling
- IBM Series/1 Realtime Programming System Indexed Access Method
- IBM Series/1 Realtime Programming System Address Translator Transient Support
- IBM Series/1 Base Program Preparation Facilities to IBM Series/1 Program Preparation Subsystem Object Module Conversion Program
- IBM Series/1 Realtime Programming System Interactive IPL Loader

Refer to Chapters 3 through 6 for additional information about the features of these programs.

Realtime Programming System Version 2 Features

The Realtime Programming System Version 2 (5719-PC2) provides all of the facilities of Version 1 plus support for the IBM 4962 Disk Storage Unit Models 3 and 4 with a capacity of 13,962,240 bytes. This increase in disk storage permits more program and data storage on a single disk for users with large program libraries and large data files. Other features of the Realtime Programming System Version 2 are:

- Support for up to 128K bytes of physical storage
- Storage overlay support
- BSC initial program load (IPL)
- Secondary storage (above 64K bytes) assigned to a partition at SYSGEN with optional IPL override
- Write with read verify option
- Additional communication support:
  - Start/Stop (asynchronous) communications—Teletype Models ASR 33/35 or equivalent in point-to-point switched connections. (Determination of equivalency is a user responsibility.)
  - Binary synchronous communications—To another IBM Series/1 using Realtime Programming System Version 2 (5719-PC2) in point-to-point switched and nonswitched connections; to an IBM System/3 using CCP or RPG in point-to-point switched and nonswitched connections; to an IBM System/370 using DOS/VS BTAM or VTAM in point-to-point switched and nonswitched connections.
  - EXIO support for communications—This facility allows the user to access the binary synchronous and asynchronous communications features at a basic level (EXIO) as well as the physical level (Read/Write). This allows access to all facilities supported by the hardware features. The EXIO and Read/Write support are mutually exclusive for an attachment.
- Automatic device backup for printers and the teletypewriter
- IPL and dump of a remote Series/1 by a host Series/1 using BSC operations
- Use of the basic access level for communication operations
- Interactive debugging package

Compatibility

The Realtime Programming System Version 2 is upward compatible with Versions 3 and 4. Problem-state programs that compile and execute on the Realtime Programming System Version 2 will also compile and execute on Versions 3 and 4.

The Program Preparation Subsystem Version 2 is required to prepare programs to execute on the Realtime Programming System Version 2.

Although the Program Preparation Subsystem Version 2 executes on Realtime Programming System Version 2, it can also prepare programs for use on the Realtime Programming System Version 1.

Supported Licensed Programs

The Realtime Programming System Version 2 supports the following IBM Series/1 licensed programs:

- IBM Series/1 Program Preparation Subsystem Version 2 (5719-AS2)
- IBM Series/1 Realtime Programming System PL/I Compiler and Resident Library Version 1 (5719-PL1)
- IBM Series/1 Realtime Programming System PL/I Transient Library Version 1 (5719-PL3)
- IBM Series/1 Realtime Programming System FORTRAN IV Compiler and Object Support Library (5719-F01)
- IBM Series/1 FORTRAN IV Realtime Subroutine Library Versions 1 (5719-F03) and 2 (5719-F04)
- IBM Series/1 Realtime Programming System Mathematical and Functional Subroutine Library Versions 1 (5719-LM1) and 2 (5719-LM2)
- IBM Series/1 4987 Programmable Communications Subsystem Preparation Facility (5719-CS0)
- IBM Series/1 4987 Programmable Communications Subsystem Execution support (5719-CS1)

Additional programming support is provided through the following Programming RPQs:

- IBM Series/1 Realtime Programming System Basic Sort
- IBM Series/1 Realtime Programming System Remote Job Entry
- IBM Series/1 Realtime Programming System 4978 Display Station Support
- IBM Series/1 Realtime Programming System Indexed Access Method
- IBM Series/1 Realtime Programming System Disk Spooling
- IBM Series/1 Realtime Programming System Address Translator Transient Support
- IBM Series/1 Base Program Preparation Facilities to Series/1 Program Preparation Subsystem Object Module Conversion Program
- IBM Series/1 Realtime Programming System Interactive IPL Loader
- IBM Series/1 Realtime Programming System Multiple Terminal Manager
- IBM Series/1 Realtime Programming System Remote Management Utility
- IBM Series/1 Realtime Programming System Transient Activity Tool

Refer to Chapters 3 through 6 for additional information about the features of these programs.

Realtime Programming System Version 3 Features

The Realtime Programming System Version 3 (5719-PC3) provides all of the facilities of Version 2 plus a multiple address space management environment. Multiple address space management provides more isolation of individual programs. It allows flexible usage of up to 256K bytes of processor storage. Some of the features of multiple address space management are:

- Support for up to 256K bytes of storage.
- Up to 15 user partitions.
- Partitions of up to 64K bytes of primary storage, with up to 256K bytes of secondary storage.
- Dynamic partitions that are created upon demand.
- Relocatable task sets, which can be executed in a partition or at an origin other than the one for which it is built.
- Multiple-address space partitions that allow a shared task set to be shared system-wide.
- System-wide events and queues in the shared task set.
- Separation of system instructions and data into two address spaces (through a SYSGEN option) so that the system partition can exceed 64K bytes; this is called an instruction/data (I/D) split system.

Compatibility

The Realtime Programming System Version 3 is upward compatible with Version 4. Problem state source programs which compile and execute on the Realtime Programming System Version 1 or Version 2 may require source modifications to compile and execute on Version 3.

The Program Preparation Subsystem Version 3 is required to prepare programs to execute on Version 3 of the Realtime Programming System. It can also prepare programs for use on the Realtime Programming System Version 1, Version 2, Version 3, or Version 4.

Supported Licensed Programs

The Realtime Programming System Version 3 supports the following IBM Series/1 licensed programs:

- IBM Series/1 Program Preparation Subsystem Version 3 (5719-AS3)
- IBM Series/1 PL/I Compiler and Resident Library Versions 1 (5719-PL1) and 2 (5719-PL2)
- IBM Series/1 Realtime Programming System PL/I Transient Library Versions 1 (5719-PL3) and 2 (5719-PL4)
- IBM Series/1 Realtime Programming System FORTRAN IV Compiler and Object Support Library (5719-FO1)
- IBM Series/1 Realtime Programming System FORTRAN IV Realtime Subroutine Library Versions 1 (5719-FO3) and 2 (5719-FO4)
- IBM Series/1 Realtime Programming System Mathematical and Functional Subroutine Library Versions 1 (5719-LM1) and 2 (5719-LM2)
- IBM Series/1 4987 Programmable Communications Subsystem Preparation Facility (5719-CS0)
- IBM Series/1 4987 Programmable Communications Subsystem Execution Support (5719-CS1)
- IBM Series/1—System/370 Channel Attach Program (5719-CA1)
- IBM Series/1 COBOL Compiler and Resident Library (5719-CB1)
- IBM Series/1 COBOL Transient Library (5719-CB2)
- IBM Series/1 Sort/Merge (5719-SM1)
- IBM Series/1 Indexed Access Method (5719-AM1)

Additional programming support is provided through the following Programming RPQs:

- IBM Series/1 Realtime Programming System Basic Sort
- IBM Series/1 Realtime Programming System Remote Job Entry
- IBM Series/1 Realtime Programming System 4978 Display Station Support
- IBM Series/1 Realtime Programming System Disk Spooling
- IBM Series/1 Realtime Programming System Indexed Access Method
- IBM Series/1 Realtime Programming System Address Translator Transient Support
- IBM Series/1 Base Program Preparation Facilities to IBM Series/1 Program Preparation Subsystem Object Module Conversion Program
- IBM Series/1 Realtime Programming System Interactive IPL Loader
- IBM Series/1 Realtime Programming System/Multiple Terminal Manager
- IBM Series/1 Realtime Programming System Remote Management Utility
- IBM Series/1 Realtime Programming System Transient Activity Tool

Refer to Chapters 3 through 6 for additional information about the features of these programs.

Realtime Programming System Version 4 Features

The Realtime Programming System Version 4 (5719-PC4) provides all of the facilities of Versions 1 through 3 plus support for:

- Command language facility—consists of an initialization task set, a terminal handler, a command language interpreter with its own command language, and a set of IBM-supplied commands written in the command language. The initialization task set is used to customize the command language facility to support the I/O devices within the user's hardware configuration, especially the terminals.

  The terminal handler is a task set which executes in a partition. It provides the ability to start and stop the interpreter from the user's terminal as different programmers desire to use the facility.

  The interpreter is a task set which executes in a partition. The interpreter is activated either by the terminal handler as the result of a LOGON command entered at the user's terminal, or by entering a TSET STR operator command at the operator station. When activated, the interpreter requests the user to enter his 'ID'. At this time, a complete set of program development data sets are made available for his use. Each concurrent user needs a terminal and a partition (for the interpreter) dedicated to his interactive session. Commands entered at the user's terminal cause the interpreter to locate the corresponding command file and execute it. The supplied command files provide functions which can be classified into the following categories:
  - Volume and data set allocation and deletion
  - DSD definition
  - Data transfer
  - Data backup and recovery
  - Data display or printout
  - Directory report
  - Program preparation
  - Program execution

  This facility provides:
  - Online program development through simplified commands
  - Commands processed by realtime interpreter that invokes requested services
  - Ease-of-use, reduced pre-planning and system knowledge through prompting and tutorial support, and help function for commands and predefined program development data set structure
  - Increased productivity through minimum key strokes, user-sensitive command syntax, and effective defaults
  - Interactive session with multiple concurrent users
  - Ability to execute a given task set in multiple partitions at the same time
  - Supplied set of commands may be supplemented by user-written command files written in the command language

  The following (if installed) can be invoked through the Command Language Facility:
  - Program Preparation Subsystem text editor, assembler and application builder
  - PL/I, COBOL, and FORTRAN IV compilers •

- SNA Support—provides a Data Flow Control level interface for support as a multiple Logical Unit Cluster Controller on an SNA network controlled by an IBM System/370 with Operating System/Virtual Storage 2 (OS/VS2) with Multiple Virtual Storage (MVS), or Single Virtual Storage (SVS) and ACF/VTAM or ACF/TCAM. Series/1 SNA support is also compatible with IMS/VS Version 1 using OS/VS2. See Figure 2-5.

SNA support category	Functions
User interface	User interface for SNA support consists of system generation facilities to allow physical network definition, and execution time function to allow for *connection to or disconnection from* the network and the *sending* and *receiving* of messages.
System definition	Defining and tailoring the Realtime Programming System SNA support is done under SYSGEN control. If the user requests this support during the SYSGEN operation, the ACTIVATE NETWORK/DEACTIVATE NETWORK commands are automatically included in the Command Definition Table. The user is also prompted with additional questions concerning the network, the logical unit definition, and the line definition.
Network attachment	Network attachment is the ability to connect a Series/1 system, by way of the Network Control Program (ACF/NCP/VS), to an SNA network. The processes provided to support this function are called network activation and network deactivation. Network activation loads the Realtime Programming System SNA support and connects the Series/1 to the SNA network. This establishes a means of communication between the two. Network deactivation is the process of requesting the System Services Control Point (SSCP) to initiate action causing termination of all sessions and disconnection from the network. Network deactivation also unloads the Realtime Programming System SNA support.
Session activation/deactivation	Session activation is the process of establishing a logical-unit-to-logical-unit communication path for subsequent data and control command exchange following a network attachment. This process can be requested by either a host System/370 user program or a Series/1 user program. Session deactivation is the process of terminating the logical-unit-to-logical- unit connection.
Message exchange	The message exchange service facility allows the exchange of data and commands between a Series/1 user program and the session partner (System/370 user program).
Architecture definition	• Physical unit type 2 • Multiple logical units • Transmission Subsystem profile 3 and 4 • Function management profile 3 and 4 • Data flow control level interface

Figure 2-5. SNA support functions

- Screen Formatter support—the screen formatter provides screen design and presentation support. It provides a consistent interface to FORTRAN IV, COBOL, and PL/I applications by providing display device transparency. The Screen Formatter supports 4978/4979 Display Stations and the 5250 Information Display System (refer to Chapter 5 for additional information). The services it provides are:
  - Creates screen formats
  - Defines amount and type of field processing
  - Defines data structure maps
  - Verifies/changes screen formats
  - Verifies and manages data as defined by maps created during screen definition
- SYSGEN enhancements—Version 4 SYSGEN enhancements reduce the number of steps required prior to system generation by the automatic creation of work files, spec files, SYSRES volume, DSDs, control block requirements, and rollin/rollout requirements.
- Dynamic Transient Pool Management—the Dynamic Transient Pool Manager controls and maintains a pool of most frequent, recently-used system transients in unaddressable physical storage. This allows the user to generate a heavily transient system and still have high performance. This support is optional and only available in the multiple address space environment.
- Single/multiple address space support—Version 4, through system generation options, makes it possible to build either a single address space system or a multiple address space system environment. New devices and communication functions will be supported in either environment. This support provides relocatable task sets for both single and multiple address space system environments. User task sets may be built by the Program Preparation Subsystem Version 3 or 4 to execute in either environment.
- Additional System/Stand-Alone Utilities—additional system utilities provide facilities for obtaining a formatted report of the contents of a DSD table. Stand-alone disk-diskette Save/Restore facilities are also available for backup for the following device combinations:
  - IBM 4363 Disk Subsystem/IBM 4966 Diskette Magazine Unit
  - IBM 4962 Disk Storage Unit Models 1F, 2F/IBM 4964 Diskette Unit
- External DSD Tables—the use of an external DSD table may be optionally specified at task set execution time. An external DSD table resides outside of a task set library volume on any direct access storage device as any named data set or member of a partitioned data set. This facility allows multiple copies of the same task set to be executed concurrently, each with a different DSD table in effect.
- Communications terminal support:
  - IBM 2740 Communications Terminal (Models 1 and 2) in multipoint connections
  - IBM 2741 Communications Terminal in point-to-point (switched and nonswitched) connections
  - Teletype Models ASR 33/35 Data Terminals or equivalent in point-to-point (switched or nonswitched) connections

- Binary Synchronous communications support:
  - IBM 3271 Control Unit (Models 1 and 2) with attached 3277, 3284, 3286, and 3288 terminal on a multipoint line
  - IBM 3275 Display Station (Models 1 and 2) on switched point-to-point or multipoint line
  - IBM 6670 Information Distributor with BSC feature on a switched or nonswitched point-to-point line
  - IBM System/370 OS/VS1, OS/VS2, IMS/VS (BTAM only) in point-to-point (switched and nonswitched) connections
  - IBM System/370 OS/VS1 BTAM, Series/1 as a multipoint tributary
- BSC and Start/Stop support:
  - Multipoint via device handler
    - —As a control station
    - —As a tributary station (BSC only)
    - —Autopoll
    - —Poll list modification and maintenance
  - Online testing
    - —BSC device including 3270 Information Display Systems
    - —Start/stop (2740 Models 1 and 2 and 2741 only)
  - Expanded DEFINE utilities
- 4963 Support—the 4963 Disk Subsystem Attachment allows up to four disk storage units to be attached to a Series/1 computer. These multiple disk units are nonremovable, magnetic disks having the capacity of from 23 million to 64 million bytes.
- 4966 Support—the 4966 Diskette Magazine Unit allows random selection of diskettes having singleordouble-density recording. Selection can be made from one of three single diskettes or the two removable magazines. Each magazine can contain a maximum of ten diskettes, each diskette having a data capacity of 1.2 million bytes.
- Generic System Residence Device Names—the device name DISK is a generic name for the device from which IPL occurs and from which the system task set is loaded into storage. All DSD references to the device name DISK will go to the system resident device automatically.
- Dynamic Control Blocks—the system will dynamically allocate the necessary control blocks as required during execution of an application. For performance reasons, control blocks may still be predefined at application build time in user task sets.
- Dynamic Device Configuration —additional devices, of like type, can dynamically be added online to an already operational system. This feature supports most IBM devices.
- Expanded diskette data management—double density (4966 only), as well as single density, diskettes are supported. A 512-byte physical sector size is also supported for nonsystem formatted diskette data sets at all levels of access. A 1,024-byte physical sector size is supported at the EXIO level only. Volume sequencing support is also available for multivolume diskette data set processing.

- Global DSD Support—the system search order for a given DSD is the following:
  - DSDT of the appropriate executing task set
  - DSDT of the currently active shared task set, if applicable
  - DSDT of the active system task set
- DSD DEV=DUMMY Support—data set and device management services will provide successful return codes without actually performing the requested I/O access functions if the generic device name, DUMMY, is specified in the corresponding DSD. This allows application programs to be tested without the need to patch out or remove I/O access calls to the system for devices which may be unavailable or inoperative.

Compatibility

Problem state source programs that compile and execute on the Realtime Programming System Version 3 will also compile and execute on Version 4. Problem state source programs that compile and execute on the Realtime Programming System Versions 1 or 2 may require minor source modifications to compile and execute on Version 4.

The Program Preparation Subsystem Version 3 or 4 is required to prepare programs to execute on the Realtime Programming System Version 4.

Supported Licensed Programs

The Realtime Programming System Version 4 supports the following IBM Series/1 licensed programs:

- IBM Series/1 Program Preparation Subsystem Version 4 (5719-AS4)
- IBM Series/1 Program Preparation Subsystem Version 3 (5719-AS3)
- IBM Series/1 Realtime Programming System PL/I Compiler and Resident Library Versions 1 (5719-PL1) and 2 (5719-PL2)
- IBM Series/1 Realtime Programming System PL/I Transient Library Versions 1 (5719-PL3) and 2 (5719-PL4)
- IBM Series/1 Realtime Programming System FORTRAN IV Compiler and Object Support Library Version 1 (5719-FO1) and 2 (5719-FO2)
- IBM Series/1 FORTRAN IV Realtime Subroutine Library Versions 1 (5719-FO3) and 2 (5719-FO4)
- IBM Series/1 Realtime Programming System Mathematical and Functional Subroutine Library Versions 1 (5719-LM1) and (5719-LM2)
- IBM Series/1 4987 Programmable Communications Subsystem Preparation Facility (5719-CS0)
- IBM Series/1 4987 Programmable Communications Subsystem Execution Support (5719-CS1)
- IBM Series/1 4987 Programmable Communications Subsystem Extended Execution Support (5719-CS2)
- IBM Series/1—System/370 Channel Attach Program (5719-CA1)
- IBM Series/1 COBOL Compiler and Resident Library (5719-CB1)
- IBM Series/1 COBOL Transient Library (5719-CB2)
- IBM Series/1 Sort/Merge (5719-SM1)
- IBM Series/1 4969 Magnetic Tape Subsystem Support (5719-TA4)
- IBM Series/1 5250 Information Display System Attachment Support (5719-TA1)
- IBM Series/1 Indexed Access Method (5719-AM1)
- IBM Series/1 Realtime Programming System Screen Formatter (5719-SF1 and 5719-SF2)

Additional programming support is provided through the following Programming RPQs:

- IBM Series/1 Realtime Programming System Basic Sort
- IBM Series/1 Realtime Programming System Remote Job Entry
- IBM Series/1 Realtime Programming System 4978 Display Station Support
- IBM Series/1 Realtime Programming System Disk Spooling
- IBM Series/1 Base Program Preparation Facilities to Series/1 Program Preparation Subsystem Object Module Conversion Program
- IBM Series/1 Realtime Programming System Interactive IPL Loader
- IBM Series/1 Realtime Programming System Multiple Terminal Manager
- Remote IBM Series/1 Realtime Programming System Management Utility
- IBM Series/1 Realtime Programming System Transient Activity Tool

Refer to Chapters 3 through 6 for additional information about the features of these programs.

System Requirements

The minimum system requirements to support system generation of the Realtime Programming System Versions 1 through 4 are:

- Processor
  - For Version 1 or Version 2 an IBM 4953 or 4955 with at least 48K bytes of processor storage. For Version 3 or Version 4 an IBM 4952 Processor is also supported. Storage increments of 32K are available on the 4952.
  - For a Version 3 or Version 4 multiple address space management environment: an IBM 4955 Processor Model B or D with the Storage Address Relocation Translator feature or a 4955 Processor Model E with at least 64K bytes of processor storage.
  - For a Version 4 single address space management environment: an IBM 4952, 4953, or 4955 Processor is supported with at least 64K bytes of processor storage.
- Program Preparation—a 64K-byte minimum system with at least a 16K-byte partition is required for the Program Preparation Subsystem to operate in a multiple address space management environment, or a 48K-byte minimum system with at least an 18K-byte partition to operate in a single address space management environment.
- Operator Station—either an IBM 4979 Display Station or a Teletype Model ASR 33/35 or equivalent ASCII device.

  Note: If a teletypewriter or ASCII device is to be used as the system console, the processor must be equipped with the Teletypewriter Adapter Feature (7850).
- Disk/Diskette—one IBM 4962 Disk Storage Unit Model 2 or 2F (combination disk/diskette unit), or one IBM 4962 Disk Storage Unit Model 1 or 1F and one IBM 4964 Diskette Unit, or one IBM 4963 Disk Subsystem. (The IBM 4963 and 4966 are supported only by Version 4.)
- Printer—either the IBM 4973 Line Printer or the IBM 4974 Printer.

  Note: The operating system supports multiple devices of the same type mentioned in the preceding list (except the processor).

The optional hardware that is available for use with the Realtime Programming System is:

- 4969 Magnetic Tape Subsystem
- 4963 Disk Subsystem
- 4966 Diskette Magazine Unit
- The IBM 4982 Sensor Input/Output Unit, which supports:
  - Analog input
  - Analog output
  - Digital input
  - Digital output
  - Process interrupts
- Integrated Digital Input/Output Nonisolated (#1560)
- Timers (#7840)
- Teletypewriter adapter feature (#7850)
- Communications features for start-stop and binary synchronous communications
  - Asynchronous Communications Single Line Control (#1610)
  - Binary Synchronous Communications Single Line Control (#2074)

    Note: The Binary Synchronous Communications Single Line Control (#2074), with an IPL jumper installed, can be used as an IPL device.
  - Binary Synchronous Communications Single Line Control/High Speed (#2075)
  - Asynchronous Communications 8-Line Control (#2091)
  - Asynchronous Communications 4-Line Adapter (#2092)
  - Binary Synchronous Communications 8-Line Control (#2093)
  - Binary Synchronous Communications 4-Line Adapter (#2094)
  - Communications Indicator Panel (#2000)
  - Feature Programmable Multitime Controller (8 line) (#2095)
  - Feature Programmable Multiline Adapter (4 line) (#2096)
- Floating-point (#3920)
- Programmer console (#5650)
- Storage Address Relocation Translator (#6335)
- IBM 4999 Battery Backup Unit

To support BSC IPL and BSC dump, the host Series/1 system must have any of the BSC communications features (2074, 2075, 2093/2094). The remote system must have one of the BSC single line features (2074, 2075).

Device and feature performance is dependent upon hardware configuration, Realtime Programming System generation options, and application program design.

Program Preparation Subsystem

The Program Preparation Subsystem, Versions 1 through 4, is a set of programs that offers:

- A general-purpose job stream processor for handling batch processing activity
- Program preparation facilities for creating realtime and batch applications
- Rapid installation and ease-of-use features

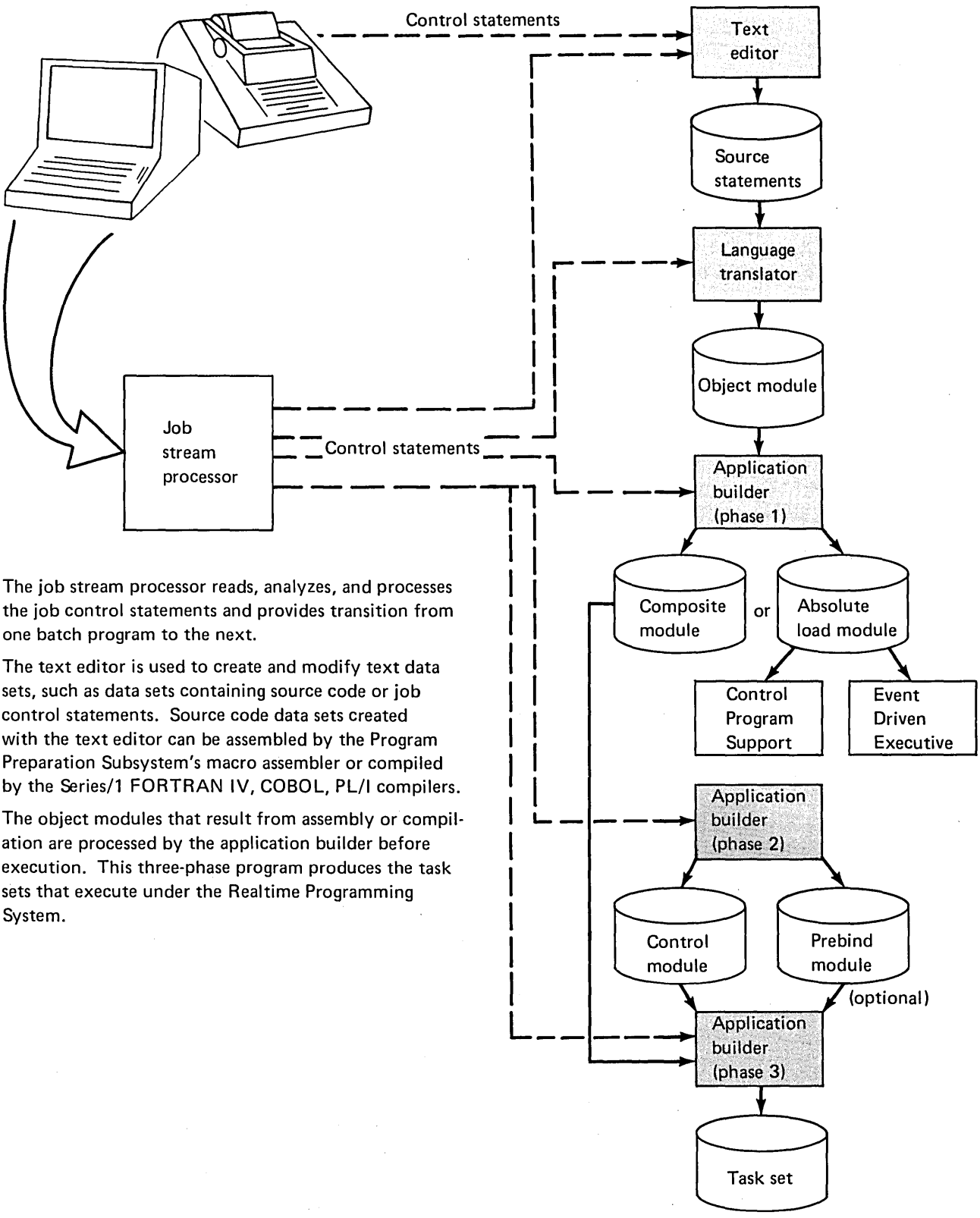
The subsystem provides minimal installation requirements and ease of use by delivering the subsystem in a ready-to-use condition. A part of the delivered package is a pre-built subsystem. Default values for all variables have been chosen so that the pre-built subsystem meets the needs of many users without further tailoring.

The subsystem programs can run concurrently with realtime applications under the realtime supervisor or, in the absence of realtime applications, can run under the realtime supervisor in a simple batch environment. In Version 4, the subsystem facilities can be executed under the command language facility. Batch processing provides a convenient method of invoking and executing programs, with little or no operator intervention required.

The text following Figure 2-6 introduces the features of each of the Program Preparation Subsystem components:

- Job stream processor
- Text editor
- Macro assembler
- Application builder
- Macro preprocessor (Version 4 only)

Preparing Programs for Execution. The basic sequence of steps to prepare programs for execution under the Realtime Programming System supervisor are illustrated in Figure 2-6. To prepare programs for execution under Control Program Support or the Event Driven Executive operating System, you follow the sequence of steps as shown in Figure 2-6, with one exception. An absolute load module is created, rather than a task set.



The job stream processor reads, analyzes, and processes the job control statements and provides transition from one batch program to the next.

The text editor is used to create and modify text data sets, such as data sets containing source code or job control statements. Source code data sets created with the text editor can be assembled by the Program Preparation Subsystem's macro assembler or compiled by the Series/1 FORTRAN IV, COBOL, PL/I compilers.

The object modules that result from assembly or compilation are processed by the application builder before execution. This three-phase program produces the task sets that execute under the Realtime Programming System.

Figure 2-6. Program preparation sequence

Job Stream Processor

The job stream processor controls batch processing activity. It reads and analyzes the stream of job control statements that specify the task sets to be executed and the data sets and devices that the task sets use. Control statements and data that apply to the execution of a task set are grouped into steps; related steps, such as an assembly and application build, are grouped into jobs. The job stream processor provides transition from step to step and from job to job.

Control statements can come from a variety of sources and can be entered through an interactive device, including the operator station, or the job stream processor can read them from a disk or diskette data set. During a batch processing session, the source of control statements can be changed from one device or data set to another.

The ability to specify the data sets and devices that a task set will use by coding job control statements gives flexibility. A task set can be device-independent because the data sets and devices it uses can be changed by simply changing job control statements. For example, input data for a program can come from a disk in one job, an interactive device in another, and a diskette in a third.

Control statements that specify data sets and devices can be grouped into lists and stored on the system.

Job control statements provide capabilities to do the following:

- Identify a task set to be executed
- Define data sets and devices; the user can create or delete permanent data sets and can create temporary data sets
- Pass parameters to a task set
- Change the source of job control statements during a batch processing session
- Assemble or compile a program, build a task set, and execute the task set by coding job control statements
- Include data with control statements
- Delimit jobs and steps
- Cancel steps in order to correct control statement errors
- Include comments on and between control statements

Detailed information about the job stream processor is contained in the *Batch User's Guide*.

Text Editor

The text editor is used to create, modify, list, get, and save text data sets. These data sets can be used as input to the subsystem's macro assembler or to a compiler.

The text editor executes in the batch partition. Once invoked, the text editor can be used interactively by entering commands and text from an interactive device, or it may be used in a noninteractive mode if commands and text come from a data set.

The editor's input can come from the IBM 4979 Display Station. If the display station is also used as the interface to the operating system, the text editor operates in shared (split-screen) mode; the user can define part of the screen for system use, while the remaining screen area is used by the editor.

The text editor's commands provide the capability to do the following:

- Copy or move one or more lines from one location to another
- Change or replace text within a field for one or more lines
- Delete one or more or all lines
- List or display all or part of a single line or of multiple lines
- Search for text and print each line containing the test
- Insert lines
- Save text in and retrieve text from a disk or diskette data set
- Set tabs
- Clear the editor workspace
- Display the current settings for session variables: that is, line length, last line number, tab character, tab columns, portion of the line to be displayed, and number of records in the workspace
- Suspend an editing session and resume it without respecifying session variables and with the same workspace conditions that existed in the prior session
- Invoke the Macro Preprocessor (Version 4 only)

Detailed information about the text editor is contained in the *Text Editor User's Guide*.

Macro Assembler

The macro assembler translates symbolic source statements into an object module, which consists of object code and information that the application builder uses for its processing. The macro assembler executes as a batch job, either alone or as part of a multi-step job that can assemble, build, and execute a task set.

The assembler processes three types of source statements: machine instructions, assembler instructions, and macro instructions. Each machine instruction is the mnemonic representation of a single processor instruction. Assembler instructions direct the assembler to perform certain operations such as defining data constants or reserving storage areas. Macro instructions generate a predefined sequence of machine and assembler instructions.

Assembler Instructions. Assembler instructions perform these functions:

- Define and structure control sections, dummy control sections, common control sections, and global control sections
- Control base register usage
- Adjust the location counter
- Define data or reserve storage for data
- Maintain a control section stack
- Define entry points
- Identify external symbols and weak external symbols
- Assign values to symbols and reference registers symbolically
- Copy source statements from a library
- Communicate between subroutines through parameters
- Control listing format and content
- Change the start, end, and continuation columns for source statements
- Check sequence of source statements

Macro Instructions. The macro instructions perform these functions in a macro definition:

- Branch conditionally and unconditionally, which allows conditional assembly
- Define and initialize global variables; that is, variables used to communicate between macro definitions
- Define and initialize local variables; that is, variables used within a macro definition
- Assign values to global and local variables; variables can contain arithmetic, binary, or character data
- Generate error messages and comments

Assembler Options. Assembler options are specified on a job control statement. Options are available to do the following:

- Request or suppress listings
- Control the content and format of listings
- Request or suppress macro phase processing
- Dump the assembler's internal work data sets
- Request or suppress object module output

For detailed information about the assembler, refer to the *Macro Assembler User's Guide*. For information about the supervisor, data management, and communications macros provided by the Realtime Programming System, refer to the *Macro Reference* manual.

Application Builder

Detailed information about how to use the application builder to build task sets is contained in the *Application Builder User's Guide*. The application builder creates a task set from object modules produced by the macro assembler or a compiler and from information specified on application builder control statements. The task set contains programs, data, and control blocks, including a control module and an optional prebind module. A control module is a set of tables and control blocks that contain control and parameter information pertaining to the task set. This information is used by the operating system to execute requested functions. The prebind module contains specifications used during task set installation, a process that enables a task set to start execution more rapidly.

The application builder is a three-phase program. All phases can run as a single job or step, or the phases can be executed in multiple jobs or steps.

Phase 1 Processing. Phase 1 can combine object modules to create absolute load modules. Phase 1 also combines object modules into composite modules, which are further processed by phase 3. Composite modules can be simple or overlay structured. A simple structure contains a resident segment (one or more object modules) that remains in primary storage for the duration of task set execution. Overlay structures contain overlay (disk and storage) segments in addition to the resident segment.

Phase 2 Processing. Phase 2 creates the control module and the optional prebind module according to information supplied on phase 2 control statements.

Phase 3 Processing. Phase 3 creates task sets by:

- Identifying the task set as a user task set or a shared task set
- Optionally providing resolution of external references to a previously built shared task set or to an automatic call library
- Providing resolution of external references from the optional prebind module and produces a resolved prebind module in the task set library
- Combining composite modules and resolving addresses
- Resolving external references
- Combining global sections into a task set global area
- Combining overlay (disk and storage) segments into an overlay data set
- Combining resident segments, their associated common and overlay areas, and the control module with addresses assigned and references resolved

At the user's request, the application builder produces maps of the composite module and the task set.

Macro Preprocessor (Version 4 only)

The macro preprocessor is the Program Preparation Subsystem Version 4 component that provides a facility for creating preprocessed macro programs. The preprocessor is a task set that is invoked through the job stream processor and requires only a few control statements to initiate the preprocessing procedure.

A preprocessed macro program is an executable load module that performs the same functions as its corresponding macro in generating code and MNOTEs and in updating GLOBAL values. Once a preprocessed macro program is prepared and resides in a preprocessed macro library, the assembler can use the preprocessed macro program as an overlay to more efficiently handle the corresponding macro calls. The preprocessor creates the preprocessed macro by first converting the macro to a source program, which is then assembled to create an object module. The object module is then processed by the application builder to create the preprocessed macro program.

Program Preparation Subsystem Version 1 Features

The Program Preparation Subsystem Version 1 (5719-AS1) consists of four components—a job stream processor, text editor, macro assembler, and application builder.

- Job stream processor facilities allow the user to:
  - Assign I/O units and data set to batch jobs
  - Create temporary and permanent data sets
  - Predefine and store lists specifying the data set and device environments for particular jobs
  - Override predefined environments from the job stream
  - Group batch work into jobs containing dependent execution steps
  - Pass parameters from the job stream to batch programs
  - Redirect the source of the job stream from one device to another during processing
  - Perform compile, load, and go operations with a minimum of effort
- Text editor facilities allow the user to:
  - Replace text data within a field
  - Copy lines of text from one area to another
  - Change a character string in one or more lines
  - Delete one or more lines
  - Search for text and print each line containing the text
  - Insert new lines of text
  - List one or more selected lines
  - Move lines of text from one area to another
  - Store a created data set on disk or diskette
  - Retrieve a specified data set from disk or diskette
- The macro assembler provides:
  - A function-oriented assembler language for specifying machine instructions
  - A macro language facility
  - Conditional assembly capability within macros
  - Sectional assembly capability
  - Assembler options for listing control
  - Relocatable object module output
  - Listing output that can include the source program and object text, external symbol dictionary, relocation dictionary, cross-reference table, error messages, and statistics
- The application builder provides:
  - The ability to change performance-impacting variables such as priorities, task scheduling, and queue space allocation without reassembly
  - Specification of resident and nonresident code
  - Optional selection of printed output such as control statement listing, control section names and addresses, a task set map, and diagnostic listings

Compatibility

The Program Preparation Subsystem Version 1 has the ability to prepare applications that will execute on the Realtime Programming System Version 1 and Control Program Support. The Program Preparation Subsystem Version 1 requires the same minimum system configuration required by the Realtime Programming System Version 1. The Program Preparation Subsystem Version 1 requires at least a 16K-byte partition.

Program Preparation Subsystem Version 2 Features

The Program Preparation Subsystem Version 2 (5719-AS2) provides all the facilities of Version 1 plus support for:

- Overlay manager controlling both disk and storage overlays
- New job stream processor functions:
  - Restart the job stream processor after unrecoverable errors
  - Cancel a currently executing step or a job that has not yet been processed
  - Display the status of batch partition activity—the job, step, and task set currently executing
- New text editor commands:
  - Replace one or more consecutive lines with another text line or lines
  - List all text editor commands and the syntax for each command
  - Display the syntax for any command
  - Edit one page at a time from the display station
  - Full screen editing mode
  - Create, change, or delete data set definition (DSD) statements
  - Submit an input stream to be queued to the job stream processor
- Storage overlays by the application builder

Compatibility

The Program Preparation Subsystem Version 2 has the ability to prepare applications that will execute on the Realtime Programming System Version 1 and Version 2. The Program Preparation Subsystem Version 2 requires the same minimum system configuration required by the Realtime Programming System Version 2. The Program Preparation Subsystem requires at least a 16K-byte partition.

Program Preparation Subsystem Version 3 Features

The Program Preparation Subsystem Version 3 (5719-AS3) provides all the facilities of Version 2 plus it has the ability to build task sets to execute in a Multiple Address Space Management environment provided by the Realtime Programming System Version 3 and Version 4 or a Single Address Space Management environment provided by the Realtime Programming System Version 4. Version 3 usability enhancements are:

- Automatic creation of data set definitions for data sets used by the application builder
- Automatic deletion of an existing data set with the same name as one being defined (if data set is too small)
- Optional update of the system data set definition table for newly built task sets
- Ability to build Relocatable task sets

Compatibility

The Program Preparation Subsystem Version 3 has the ability to prepare applications that will execute on the Realtime Programming System Version 1, Version 2, Version 3, and Version 4.* The Program Preparation Subsystem Version 3 requires the same minimum system configuration required by the Realtime Programming System Version 3 or 4.

The storage size of the minimum configuration, 64K bytes, allows at least a 16K-byte partition for the Program Preparation Subsystem Version 3 when executing under the multiple address space management environment of the Realtime Programming System Version 3 or 4.

A 48K-byte system with at least an 18K-byte partition is required when executing under the single address space management environment of the Realtime Programming System Version 4.

* If IBM Series/1 Magnetic Tape Subsystem applications are to be prepared, the Program Preparation subsystem Version 4 is a requirement.

Program Preparation Subsystem Version 4 Features

The Program Preparation Subsystem Version 4 (5719-AS4) provides all the facilities of Version 3 plus a new component, a *macro preprocessor*, which allows the user to develop preprocessed macro programs to significantly improve assembler performance.

In addition, the following new assembler functions are provided:

- Automatic creation of assembler work DSDs and object module output data sets
- Multiple assembly capability for processing all members of a partitioned data set or all data sets in a volume
- Support for 4952 Processor instructions

Compatibility

The Program Preparation Subsystem Version 4 has the ability to prepare applications that will execute on the Realtime Programming System Version 1, Version 2, Version 3, and Version 4. The Program Preparation Subsystem Version 4 requires the same minimum system configuration required by the Realtime Programming System Version 4.

The storage size of the minimum configuration, 64K bytes, allows at least a 16K-byte partition for the Program Preparation Subsystem Version 4 when executing under the multiple address space management environment of the Realtime Programming System Version 3 or 4.

A 48K-byte system with at least an 18K-byte partition when executing under the single address space management environment of the Realtime Programming System Version 4.

Note: A partition size of 26K bytes or greater is required when using the macro preprocessor in a multiple address space management environment. A partition size of 28K bytes or greater is required when using a single address space management environment.

Chapter 3. High-Level Languages and Supporting Libraries

This chapter introduces the IBM Series/1 high-level languages and supporting libraries available under the Realtime Programming System and Event Driven Executive operating system. They are:

- PL/I
- COBOL
- FORTRAN IV
- FORTRAN IV Realtime Subroutine Library (Realtime Programming System only)
- Mathematical and Functional Subroutine Library

PL/I Offerings

There are three separate and distinct PL/I offerings for the IBM Series/1. Each offering consists of a compiler, resident library, and transient library. Two of the offerings execute under control of the Realtime Programming System and the third executes under control of the Event Driven Executive.

- **Realtime Programming System PL/I Version 1** (Program Numbers 5719-PL1 and 5719-PL3) operates under the Realtime Programming System, Versions 1 through 4. The PL/I language supported is a subset of American National Standards Institute (ANSI) PL/I, plus extensions for sensor I/O and multitasking.
- **Realtime Programming System PL/I Version 2** (Program Numbers 5719-PL2 and 5719-PL4) operates under the Realtime Programming System, Versions 3 and 4. The PL/I language supported is the same as that supported by PL/I Version 1, above, plus these additional functions:
  - Communications support for BSC and start/stop
  - Full-screen support for 4978, 4979, and 5250 Display Stations
  - Indexed Access Method support
  - Object code optimization
  - Dynamic allocation and freeing of storage
  - Magnetic tape support
  - Sort/Merge support
  - More built-in functions and data attributes
- **Event Driven Executive PL/I** (Program Numbers 5719-PL5 and 5719-PL6) operates under the Event Driven Executive. The PL/I language supported is a subset of that implemented by PL/I Version 2 for the Realtime Programming System, as described above.

Realtime Programming System PL/I Version 1

IBM Series/1 Realtime Programming System PL/I is a problem-oriented, high-level language that can be used for programming realtime, scientific, problem-solving, and traditional data processing applications, as well as advanced applications such as transaction processing and data-base handling. PL/I is aimed at speeding up application development time by making available a wide range of facilities, including error-detection and debugging aids. Programmer coding can be kept to a minimum because many attributes and options can be automatically supplied by the compiler.

IBM Series/1 Realtime Programming System PL/I Version 1 is a subset of the American National Standard Institute (ANSI) Programming Language PL/I (ANSI X3.53-1976), plus extensions. IBM Series/1 PL/I Version 1 consists of two licensed programs: a compiler with a resident library, 5719-PL1 (needed on machines used to compile and application build) and a transient library, 5719-PL3 (needed on machines used to application build and execute). PL/I Version 1 requires two additional licensed programs for its operation: the Program Preparation Subsystem and the Realtime Programming System.

The Realtime Programming System PL/I compiler produces the object code from the user's source program. The Program Preparation Subsystem combines the object code with modules from the PL/I resident library. The resulting modules (or taskset) can then be executed under the control of the Realtime Programming System. During execution, additional modules from the PL/I transient library can be loaded dynamically.

The Realtime Programming System PL/I compiler executes in a batch environment requiring at least 28K-byte partition under control of the Realtime Programming System Versions 1 or 2. Version 3 requires at least 26K plus a 2K byte control module. The Version 4 multiple address space management environment requires at least 26K plus a 2K byte control module. The Version 4 single address space management environment requires at least 28K bytes.

The Realtime Programming System PL/I Version 1 compiler operates in the same minimum hardware configuration required to install and maintain the operating system, except that the processor must have 64K bytes of storage.

The Series/1 Realtime Programming System PL/I language permits development of application programs that can be extended or changed. Highlights of PL/I include:

- Realtime language extensions
- Input/Output capability
- Multiple data types and organizations
- Data manipulation functions
- Productivity functions
- Additional functions

Realtime Language Extensions

IBM Series/1 Realtime Programming System PL/I extends the PL/I language to permit easy development of realtime applications, while retaining the basic structure of the PL/I language. To achieve this, extensions are provided in the following areas:

- Ability to schedule, execute, and control external procedures as independent parallel tasks
- Ability to schedule and execute task sets
- Support for synchronization and control of program data and flow by using EVENT variables, LOCK variables, and deadlock avoidance
- Extension of event concepts to recognize time-of-day events, events triggered by external causes (process interrupts), repetitive events, and resetting events
- Extension of PL/I record I/O to handle sensor I/O (digital and analog)

Input/Output Capability

IBM Series/1 Realtime Programming System PL/I supports both stream and record I/O. Stream I/O statements read and write data with a minimum of programming effort, because automatic formatting and conversion are provided. The following specific options are available:

- List-directed I/O. Permits the user to read or write data with automatic formatting and conversion.
- Edit-directed I/O. Provides the user with a range of format items, including picture qualifications and control, which permits generation of complex reports with a minimum of programming effort.

Record I/O statements allow more control over I/O. The following options are available:

- Sequential asynchronous I/O. This facility is available through the use of the READ, WRITE, and REWRITE statements. The EVENT options for asynchronous I/O improve execution-time performance.
- Direct I/O. This facility is available through the use of the READ, WRITE, DELETE, and REWRITE statements with the KEY option. Asynchronous direct I/O is also permitted.
- Sensor I/O. The facility for handling both sequential and random sampling of analog and digital I/O is available through the use of the READ and REWRITE statements.
- Transient files. This form of file organization allows communication of data between operating system queues using PL/I READ and WRITE statements. The PL/I program can detect and handle the empty queue situation by coding an ON-unit for the PENDING ON condition.

Multiple Data Types and Organizations

IBM Series/1 Realtime Programming System PL/I supports arithmetic, string, and program control data. Arithmetic data can be represented in either binary or decimal radix and can be either fixed or floating point. Fixed-point binary word and doubleword precisions are supported. Decimal fixed-point data can have up to 15 digit positions, with up to 127 fractional positions (scale factor). String data can be either bit or character, with fixed or variable-length attributes. Program control data can be a label, event, activation, lock, or pointer. Entry and file parameters are also supported.

PL/I data may be organized into arrays of up to 15 dimensions or in structures (hierarchical collections of data, not necessarily of the same type). A structure can also be dimensioned.

Data Manipulation Functions

IBM Series/1 Realtime Programming System PL/I supports all PL/I operators and the major data types and statements including:

- String operations, including substrings, concatenation, and general Boolean operations
- Language built-in functions, including mathematical functions, string functions, and array functions
- Structure assignment
- Automatic data conversions in expressions
- Generalized subscripting
- Full support for internal and external procedures
- Control structures including IF—THEN, IF—THEN—ELSE, DO, and DO—WHILE

Productivity Functions

Included in this category are:

- Compile-time diagnostic messages
- Compile-time listing aids
- Execution-time diagnostic messages
- User programming and control of error conditions with the PL/I ON-handling language

Additional Functions

These functions make IBM Series/1 Realtime Programming System PL/I suitable as a general application development tool:

- Storage efficiency gained by the generation of reentrant code and support for automatic storage allocation
- Program modularity and interface checking provided by the PL/I block structure and scope rules and the ENTRY attribute
- The ability to build and manipulate chained data lists and rings using the PL/I list processing support; that is, the pointer data type and based storage

Realtime Programming System PL/I Version 2

IBM Series/1 Realtime Programming System PL/I Version 2 consists of two licensed programs: a compiler with a resident library (5719-PL2) and a transient library (5719-PL4). PL/I Version 2 includes all the facilities of Version 1 plus the following additional functions summarized below and in Figure 3-1. PL/I Version 2 requires the Realtime Programming System Version 3 or 4 and the Program Preparation Subsystem Version 3 or 4.

Additional functions	Description
Data Attributes	
DEFINED	Specifies that the variable being declared is to be associated with some or all of the storage associated with the designated base variable.
FILE VARIABLE	Enhances I/O flexibility
INITIAL	Supports AUTOMATIC variables.
PICTURE	Defines the internal and external formats of character-string and numeric character data. It also specifies the editing of data.
POSITION	Specifies the beginning of the part of a string base variable with which the defined variable is to be associated.
Built-in Functions	
ADD	Returns the sum of two values.
ATANH	Returns a floating-point value that represents the inverse (arc) hyperbolic tangent of a given value.
CEIL	Returns the smallest integer greater than or equal to a given value.
COSH	Returns a floating-point value that represents the hyperbolic cosine of a given value.
DIVIDE	Returns the quotient of two values.
FLOOR	Returns the largest integer less than or equal to a given value.
MAX	Returns, from a set of two or more arguments, the value of the argument with the largest value.
MIN	Returns, from a set of two or more arguments, the value of the argument with the smallest value.
MOD	Returns the smallest value that must be subtracted from a given value (X1) to make it divisible by a given value (X2).
MULTIPLY	Returns the product of two values.
PAGENO	Returns a fixed-point binary integer of precision (15,0) representing the current page number of a specified file.
ROUND	Returns the given value rounded.
SINH	Returns a floating-point value that represents the hyperbolic sine of a given value.
SUBTRACT	Returns the difference of two values.
SUM	Returns the sum of all the elements in a given array.
TANH	Returns a floating-point value that represents the hyperbolic tangent of a given value.
TRANSLATE	Returns a string the same length as a given string where all or some of the characters may have been changed.
TRUNC	Returns an integer that is the truncated form of a given value.
VALID	Verifies that a picture variable contains a value consistent with its declaration.
VERIFY	Returns a fixed-point binary integer indicating the position in the given string (X1) of the first character that is not in the given string (X2).

Figure 3-1. PL/I Version 2 additional functions summary

Communications Support

This support permits data communication between multiple Series/1 and between the Series/1 and the System/370 by means of the PL/I record-oriented statements: READ, WRITE, and REWRITE. Binary synchronous communications is used for this support. Start-stop communications is used to support the IBM 2740 Communications Terminal Model 1 and Teletype Models ASR 33/35 or equivalent device.

Full-Screen Support

Full-screen support is provided for the 4978, 4979, 5251, and 5252 Display Stations. When working with files, the programmer can use the following features of the display screen (as appropriate for the device):

- Full screen
- Split screen
- Pre-cursor position/post-cursor position
- Highlight characters
- Blank the screen off and on
- Scatter write
- Protect specified fields
- Control the function of program-function keys
- Scrolling
- Read modify
- Tone alarm
- Functions supported through the data stream

This may require the 4978 Display Station Support Programming RPQ Version 3 (5799-TCE) or the 5250 Information Display System Attachment Support (5719-TA1) licensed program. The 5250 is supported by the Realtime Programming System only.

Indexed Access Method Support

The Indexed Access Method permits reading, writing, and updating of records in an indexed file. The indexed file is created sequentially with unique keys assigned to the records. After the file is created, either direct or sequential processing is permitted. Either Indexed Access Method licensed program or Programming RPQ is required.

Object Code Optimization

The compiler analyzes object-code logic to eliminate unnecessary instructions. This saves execution time and storage space for PL/I programs.

Dynamic Allocation and Freeing of Storage

Through the use of two statements (ALLOCATE and FREE), the programmer can allocate and free storage dynamically during the execution of a PL/I program.

Magnetic Tape Support

The PL/I programmer can sequentially access data stored on the magnetic tape subsystem using the IBM Series/1 Realtime Programming System 4969 Magnetic Tape Subsystem Support licensed program (5719-TA4).

Sort/Merge Support

Functions available with the Sort/Merge program are supported by PL/I via the CALL PLISRT interface for PL/I programmers. This requires the Sort/Merge licensed program.

Event Driven Executive PL/I

IBM Series/1 Event Driven Executive PL/I consists of two licensed programs: a compiler with a resident library (5719-PL5) and a transient library (5719-PL6). Event Driven Executive PL/I includes all of the Realtime Programming System PL/I Version 2 language functions except:

- 5251/5252 Display Station support
- Sensor I/O
- Transient files
- Time and process interrupt events
- Delayed scheduling of tasks and programs

Realtime Programming System COBOL

IBM Series/1 Realtime Programming System COBOL is a high-level programming language oriented toward commercial applications. Series/1 COBOL allows its users to construct, compile, debug, and execute COBOL programs all on Series/1 hardware. It generates executable code for COBOL verbs and calls library subroutines that interact with the programming system and perform complex data handling. The language offers a variety of commercial features, plus facilities for handling input and output, sorting and merging data files, and structuring the source and object programs. Series/1 COBOL also includes facilities for accepting data from and displaying it at Series/1 interactive devices, and a variety of debugging and productivity aids.

IBM Series/1 Realtime Programming System COBOL consists of two licensed programs: a compiler with a resident subroutine library (5719-CB1) which translates COBOL source programs into executable machine instructions and data, and a transient subroutine library (5719-CB2) which consists of subroutines that, when needed and fetched from a data set, perform various functions for the compiled code. The transient library is required on machines where COBOL programs are to be executed, and is not required on those where only compilation is to be done.

The IBM Series/1 Realtime Programming System COBOL compiler runs under Version 3 or 4 of the Realtime Programming System and can be run in the batch partition of Version 3 or 4 of the Program Preparation Subsystem. A COBOL task set can run in the foreground or as a batch job, and it is possible to execute multiple programs concurrently. COBOL makes use of the Realtime Programming System message facility for logging run-time messages.

The COBOL compiler requires 34K bytes of main storage when operating in a Realtime Programming System environment. Execution time storage requirements depend on the characteristics of the COBOL application program. A very small program can execute in an 8K byte-partition. Typical 300 to 500 statement COBOL programs can execute in an 14K- to 18K-byte partition.

IBM Series/1 Realtime Programming System COBOL is designed according to the specifications for 1974 ANS COBOL as defined in ANSI standard X3.23-1974, as understood and interpreted by IBM as of June 1978. Except that processing of the RERUN clause is confined to checking for syntactic validity. It is designed according to the specifications for Low Intermediate Level COBOL as defined in FIPS (Federal Information Processing Standard) PUB 21-1. The product contains many useful ANS language features above the Low Intermediate Level. Series/1 COBOL Features are:

Nucleus. The nucleus contains most of the COBOL internal processing features such as arithmetic operations and program logic.

Table Handling. The table handling feature of COBOL allows the programmer to define and process fixed-length tables of up to three dimensions. The programmer can perform a sequential search to locate an item in a table. The search argument can be a subscript (which gives the ordinal position of the table entry) or an index value.

Sort/Merge. The Sort feature allows the COBOL programmer to specify that the records in one to eight data files be accepted from an input file, sorted in ascending or descending order on one or more fields, and written to an output file. The program can modify the input and output records before or after sort processing.

The Merge feature allows the programmer to merge two to eight identically sequenced files according to the ascending or descending order of a data field.

Segmentation. The Segmentation feature lets the programmer divide the Procedure Division of a COBOL program into a series of segments. The programmer can specify that some segments *(fixed segments)* must be resident in main storage while the program is running and cannot be overlaid, while the others *(independent segments)* are loaded into an overlay area when they are needed. Such a use of overlays reduces main-storage requirements during program execution.

Source Program Library. The programmer can specify that text (Configuration Section paragraphs, Input-Output Section paragraphs, FD and SD entries, record description entries, Procedure Division sections and paragraphs) be copied into a source program from a library.

Interprogram Communication. The programmer can cause transfer of control from one COBOL object program to another within a task set, and programs can exchange information. The programmer can also use the CALL statement to get access to programs written in assembler language, PL/I, and FORTRAN.

I/O Capabilities

IBM Series/1 Realtime Programming System COBOL programs can work with sequential, relative, and indexed files.

Sequential Files. A sequential file consists of records that have no keys. During creation of the file, they are written consecutively in the order in which they are presented; thereafter, they are retrieved in the same order. Sequential files can be fixed, fixed-blocked, fixed-blocked-spanned, variable, variable-blocked, and variable-blocked-spanned. For a file in variable-blocked-spanned format to be updated, the logical record size must be less than the block size (that is, a record can span only two blocks).

Relative Files. Each record in a relative file is uniquely identified by an unsigned integer value that represents the record's ordinal position in the file. A data item in each record may be designated the key for the record; that data item must not be defined in a record description entry associated with the file name. A program can access records sequentially (in the physical order they appear in the file) or randomly (by specifying the relative record number or key). A relative file must have fixed format. Each record is contained in its own block.

Indexed Files. Each record in an indexed file has a key, and access to the record is through the key value. A record description may include one or more key data items, each of which is associated with an index. Each index provides a route to the records based on the key data item associated with that index. Access can be sequential (records returned in ascending key sequence) or random (records returned in the sequence the programmer specifies).

COBOL operations on data associated with I/O Devices. Within a COBOL program, the programmer can open and close a data file, read, write, rewrite, and delete a record, accept data from any supported I/O device, and send data to any supported I/O device.

Object Program Options

The compiler allows the programmer to specify that the compiler is to produce an object module in a form suitable for the application builder. The programmer can specify that the object module is to be reentrant (if it meets certain requirements) thus allowing the same task to be used multiple times concurrently, or nonreentrant (which takes less main storage at execution).

Debugging Features

IBM Series/1 Realtime Programming System COBOL features that assist the programmer in debugging COBOL programs include: compile-time storage maps and execution-time "snapshots" of data areas; flow trace, which identifies the last statement executed before an abnormal termination; extensive error checking and error messages at five severity levels; several programmer options for controlling the form of the output listing.

Subroutine Library

The Realtime Programming System COBOL resident library and transient library routines handle:

- Arithmetic conversion
- Decimal and binary arithmetic
- Transfer of data between data areas
- File processing (communication between the object code and the operating system data management)
- Initialization and termination
- Error handling and message processing
- The Sort/Merge interface
- 4969 Magnetic Tape Subsystem support

The routines that handle conversion, arithmetic, initialization, termination, errors, and Sort/Merge, and some of the I/O routines, must be in main storage at all times during execution of a COBOL program that uses them. They make up the resident library. The remaining subroutines make up the transient library. Your installation can make the transient routines resident in main storage, thus improving performance at the expense of main-storage space; otherwise, they are maintained in a relative data set on auxiliary storage and brought in when needed.

Commonly-used library routines can be placed in the shared task set, thus permitting multiple programs or partitions to use a single copy of the routines.

Event Driven Executive COBOL

IBM Series/1 Event Driven Executive COBOL consists of two licensed programs: a compiler with a resident library (5719-CB3) and a transient library (5719-CB4). Event Driven Executive COBOL includes all of the the Realtime Programming System COBOL language functions except:

- No reentrant option
- CALL statement supports Event Driven Language programs
- No spanned records
- Compilation requires 32K bytes
- Minimum execution size for an Event Driven Executive COBOL program is 8K bytes; actual size will vary according to the number and type of source statements
- No shared task set

FORTRAN IV

IBM Series/1 FORTRAN IV is a high-level, mathematically-oriented language designed to manipulate numerical data and format input/output operations. In addition to being easily learned and understood, applications can be programmed without knowledge of the IBM Series/1 assembler language. Programming productivity is increased, since high-level languages require less coding than assembler languages.

IBM Series/1 FORTRAN IV is available for programming applications on the IBM Series/1 computer and consists of a Compiler and Object Support Library (5719-FO1 for Realtime Programming System Versions 1 through 4 and 5719-FO2 for both the Realtime Programming System Version 4 and the Event Driven Executive). The optional IBM Series/1 FORTRAN IV Realtime Subroutine Library Version 1 (5719-FO3) and Version 2 (5719-FO4) are supported by the Realtime Programming System only.

The IBM Series/1 Mathematical and Functional Subroutine Library (MFSL) Version 1 (5719-LM1) or Version 2 (5719-LM2) is required for FORTRAN IV under the Realtime Programming System. MFSL (5719-LM3) is required for FORTRAN IV under the Event Driven Executive.

The IBM Series/1 FORTRAN IV compiler produces object code. The code emphasizes compact storage and execution speed. The FORTRAN IV compiler is a serially reusable, single task set that executes as a batch job under the Program Preparation Subsystem in a Realtime Programming System environment, or in an Event Driven Executive environment. It requires at least 16K bytes of main storage. The compiler translates a source program into an object module acceptable to the application builder under the Realtime Programming System or the linkage editor under the Event Driven Executive. Source statements are analyzed by the FORTRAN IV compiler for correct syntax, and appropriate diagnostic messages are produced when errors are detected. In addition to the object module, the compiler optionally produces the following maps and listings:

- Source statement listing
- Statement label map with relative addresses
- Map of storage locations for variables and arrays
- Hexadecimal listing of the object code with statement offsets identified
- Cross-reference index for symbols and labels

The FORTRAN IV compiler can execute on an IBM Series/1 configuration that does not have floating-point support. Similarly, Realtime Programming System FORTRAN IV object programs that do not use floating-point (REAL) numbers do not require floating-point support. Event Driven Executive FORTRAN IV programs require an IBM 4955 processor with the floating-point feature (3920) for execution. The FORTRAN IV compiler operates in the same minimum hardware configuration required to install and maintain the operating system, except that at least a 16K-byte batch partition is required.

Language Elements

IBM Series/1 FORTRAN IV is a subset of American National Standard Institute (ANSI) FORTRAN, X3.9-1966, and includes the American National Standard Basic FORTRAN X3.10-1966, with the exception of object time formats, adjustable dimensions, COMPLEX data type, G-format specifications, and two-level FORMAT parenthesis. Series/1 FORTRAN IV also has many language elements not provided by ANSI Basic FORTRAN; these are referred to collectively as the Series/1 FORTRAN IV extensions.

The Series/1 FORTRAN IV extensions adapt FORTRAN to the Series/1 environment and provide greater programming flexibility. The significance of several extensions follows.

Multiple Program Support. Three FORTRAN IV statements (PROGRAM, INVOKE, and GLOBAL) permit optimal use of the interrupt scheme (INVOKE and GLOBAL are not available in Event Driven Executive FORTRAN IV). The PROGRAM statement assigns a unique name to each of several main programs, thereby establishing multiple entry points for execution in response to interrupts. The INVOKE statement selects the combination and sequence of programs to be loaded into main storage. The GLOBAL statement establishes a data area common to two or more main programs.

Logical and Relational Operations. To aid in making decisions, logical variables can be defined and assigned to true or false values. Furthermore, you can evaluate a logical or relational expression with the logical IF statement and take appropriate action according to the logical value of the expression. For example, the expression A.GT.B (A greater than B) can be evaluated to be either true or false. In this example, .GT. is a relational operator.

Bit-Level Operations. With most FORTRAN languages you work at the word or character level. To work with sensor-based activities you need to interrogate and manipulate the bits that correspond to sensor-based input/output points. Series/1 FORTRAN IV includes functions for bit-level operations. These functions, which are based on the Instrument Society of America (ISA) Standard, S61.1-1976, are:

- ICOMP or NOT — Logical *complement* of an argument
- IAND — Logical *AND* of two arguments
- IOR — Logical *OR* of two arguments
- IEOR — Logical *exclusive* of two arguments
- ISHFT — Logical *shift* of an argument
- BTEST — Logical *test* of a specified bit
- IBSET — *Set* a specified bit
- IBCLR — *Clear* a specified bit

Direct-Access Input/Output. Series/1 FORTRAN IV supports both formatted and unformatted direct-access I/O. The direct-access support is implemented through the DEFINE FILE, READ, WRITE, FIND, and FORMAT statements.

Additional READ and WRITE Parameters. The ERR= and END= parameters on the READ statement and the ERR= parameter on the WRITE statement give added flexibility and control. ERR= specifies the label of a routine to handle errors during either a sequential or direct-access READ or WRITE. END= specifies the label of a routine to handle end-of-file during a sequential READ.

Device-Independent Input/Output. Series/1 FORTRAN IV programs can be written so that assignment of physical devices to many data sets can be deferred until the program is ready to execute. These assignments can then be changed as required for subsequent use without recoding and recompiling the program.

List-Directed Input/Output. List-directed I/O simplifies data entry by freeing the programmer from FORMAT statement restrictions. Data may be entered (or is written) without regard for column or line boundaries.

Single and Double Precision. Statistical applications often require greater accuracy than single precision (7+ decimal places) can offer. The Series/1 FORTRAN IV compiler, therefore, includes a double-precision capability (16+ decimal places).

Six-Character Names. Names of up to six alphameric characters may be used for variables, arrays, functions, and subroutines within Series/1 FORTRAN IV programs. This technique allows for more meaningful names, improves program documentation, and simplifies maintenance efforts.

Expanded Character Set. The Series/1 FORTRAN IV character set includes the dollar sign ($) as an alphabetic character and the ampersand (&) and apostrophe(') as special characters in addition to the basic FORTRAN character set, which includes the letters A, B,...Z, the numbers 0,1,...9, and the characters plus (+), minus (−), slash (/), equal (=), asterisk (*), the left and right parentheses (), and the blank.

Diagnostics

IBM Series/1 FORTRAN IV provided diagnostic aids or services assist in program creation and debugging. Source statements are verified for proper syntax during compilation. For faster analysis, statement syntax can be checked without producing object code. The object support library contains routines which provide diagnostic information in case error conditions occur during program execution.

The debug facility consists of a DEBUG statement, an AT debug packet identification statement, and the TRACE statement. These statements are used to specify the desired debugging operations for a program. A log of program activity (such as a trace of routine labels) is produced while the program is executing.

Error Handling

While an IBM Series/1 FORTRAN IV program is executing, errors may result from invalid data usage or hardware failures. FORTRAN IV assists in detecting these errors and taking appropriate action. The mathematical routines check the arguments passed to them for validity. If an argument is invalid, corrective action is taken and processing continues uninterrupted. For example, if a negative argument is passed to the ALOG (natural logarithm) function, the absolute value of the argument is used. Other errors can be checked within your FORTRAN IV program. These include:

- Overflow — Arithmetic result greater than the maximum valid real number
- Underflow — Arithmetic result less than the smallest valid nonzero real number
- Divide check — Attempt to divide a real number by zero
- Function code — Illegal argument/invalid data

For I/O device errors, FORTRAN IV offers several levels of error handling. The lowest level of support is the printing of an informative message prior to program termination. If your program is to take some particular action prior to termination, an error handling routine can be specified (using the ERRXIT subroutine) to receive control when I/O errors occur. An additional level of error handling is provided by the ERR= and END= parameters on I/O statements to specify different error-handling routines for different devices and situations.

FORTRAN IV Object Support Library

The object support library is a group of subroutines designed to be combined, as needed, with the object modules produced by the compiler to form a task set executable on a Series/1 computer under control of the Realtime Programming System or Event Driven Executive supervisor. The library subroutines perform:

- Input/output processing
- Error handling
- Explicit and implicit service operations
- Bit manipulation

The compiler generates a call to the necessary library routine at the appropriate point in the object code. During application builder processing, or linkage editing, copies of these library routines are made part of the task set. Then, at execution time, the library routines perform their various functions.

FORTRAN IV Realtime Subroutine Library (Realtime Programming System only)

The IBM Series/1 FORTRAN IV Realtime Subroutine Library Version 1 (5719-FO3) and Version 2 (5719-FO4) provides optional realtime support for FORTRAN IV programs under the Realtime Programming System. These libraries consist of a group of reentrant subroutines designed to be combined, as necessary, with the object modules produced by the FORTRAN IV compiler.

Version 2 includes all the subroutines provided in Version 1, and new process I/O and realtime system interface subroutines. Figure 3-2 summarizes the functions provided in Version 1 and Version 2.

The IBM Series/1 FORTRAN IV Realtime Subroutine Library Version 1 and Version 2 require the FORTRAN IV Compiler and Object Support Library to operate.

Subroutine category	Functions
Executive function subroutines	Start and stop programs, and delay execution for a period or until a specified time.
Process I/O subroutines	Analog—read input groups sequentially or in a user-specified order, and write output points. (Version 2 only: Read outputs in a user-specified order.) Digital—read input groups, and set and/or reset digital output points or groups. (Version 2 only: Read or write outputs in user-specified order, and write outputs pulsed.)
Realtime system interface subroutines	Define and delete events and storage queues. (Version 2 only: Define and delete disk queues.) Wait for and post event completion. Add and remove an element from a queue. Define, delete, request, and release a resource. Connect to or disconnect from an interrupt. Attach and detach (terminate) a task. Queue a task set for execution and terminate it. Modify System Scheduler table and System Task Set table. Set task error exit. Read and write time-of-day. (Version 2 only: Increment time-of-day.) Set ROLLIN/ROLLOUT status. (Version 2 only: Convert EBCDIC strings to or from internal format. Load the restart supervisor. Translate code format of strings.)
Time and date subroutines	Determine the current time of day and calendar date.

Figure 3-2. FORTRAN IV Realtime Subroutine Library Versions 1 and 2 functions

Mathematical and Functional Subroutine Library

The IBM Series/1 Mathematical and Functional Subroutine Library (MFSL) Version 1 (5719-LM1) and Version 2 (5719-LM2) is a set of subroutines that aids in developing application programs under the Realtime Programming System. MFSL Version 3 (5719-LM3) is a set of subroutines, functionally equivalent to MFSL Version 2 (5719-LM2), that aids in developing application programs under the Event Driven Executive.

MFSL is required for FORTRAN IV, and the MFSL subroutines can also be used with the Series/1 assembler language or the Event Driven Language. The program preparation facilities permit the user to assemble, compile, combine, and execute programs that use the MFSL subroutines.

If floating-point operations are used, then the floating-point hardware feature (#3920) must be installed, or the floating-point emulator selected (Realtime Programming System only). If no functions that require REAL arithmetic are used, MFSL has no requirement for floating-point support in either hardware or software. The functions that require floating-point support are the following:

- Logarithmic and exponential
- Trigonometric
- Hyperbolic
- Exponentiation with REAL variables
- Arithmetic with REAL variables
- EBCDIC conversions with REAL variables

A user-written operating system that provides the required interfaces can also use the MFSL subroutines.

MFSL is compatible with any Series/1 hardware configuration that includes the processor storage and disk storage required for the MFSL subroutines used. The configuration requires floating-point support only if the user application requires REAL or floating-point arithmetic. MFSL functions that operate on integer or fixed-point variables have no internal requirements for floating-point support.

The Realtime Programming System MFSL Versions 1 and 2 include four types of subroutines: (1) mathematical functions such as SIN and SQRT, (2) EBCDIC conversion subroutines such as $ECIN (EBCDIC-to-floating-point), (3) error-checking subroutines such as FCTST (function test), and (4) service subroutines such as $FMYINT (work area initialization). Event Driven Executive MFSL only includes the first three types. Realtime Programming System MFSL Version 2 and Event Driven Executive MFSL also have commercial subroutines such as EDIT.

The library subroutines can be used in FORTRAN IV, Series/1 assembler language, or Event Driven Language programs. In FORTRAN IV, calls to the library subroutines are either at the programmer's request through explicit references to subroutine names or in response to the FORTRAN IV exponentiation notation. In Series/1 assembler language or Event Driven Language, all MFSL subroutines are invoked through explicit calls to subroutine names.

Mathematical Subroutines

The MFSL mathematical subroutines perform many commonly used mathematical operations to aid the application programmer. The explicitly called mathematical subroutines include logarithmic and exponential functions, trigonometric functions, and the miscellaneous functions of maximum and minimum values, modular arithmetic, positive difference, and transfer of sign. These subroutines are shown in Figure 3-3.

Conversion Subroutines

Input and output data conversions are made easier by the MFSL EBCDIC conversion subroutines. Numerical input data in EBCDIC format can be converted to an internal representation in integer or floating-point format. After computations in integer or floating-point arithmetic, the resulting numerical output can be converted back to an EBCDIC format. These subroutines are shown in Figure 3-3.

To use the conversion subroutines, the user program must establish input and output buffers and manage their contents by using the READ/WRITE facilities of FORTRAN IV, the Series/1 assembler language, or the Event Driven Language. The MFSL conversion subroutines always move data between a variable in the user program and an input or output buffer. Each conversion subroutine manages its buffer so that repeated calls to the subroutine will process sequential fields in the buffer. When the buffer is completely processed, the conversion subroutine goes back to the beginning of the buffer for the next conversion following a READ or WRITE I/O operation. The user call to a conversion subroutine requires a parameter list to specify the following:

- The name of the variable used in the user program
- The width (in characters) of the input or output buffer field for each conversion
- The name of the input or output buffer
- The value of a decimal scale factor (if used)
- The default number of decimal places (if used)

Error-Checking Subroutines

The MFSL subroutines communicate with the user through flags in the MFSL library work area. There are no MFSL error messages. The typical MFSL procedure for error handling is that the error-detecting subroutine (for example, SQRT detecting a negative argument) sets a flag in the library work area and then continues processing according to a predefined rule (such as taking the square root of the absolute value). To check for errors, the user must either check function arguments before invoking a subroutine or use the error-checking subroutines to validate the results. These subroutines are shown in Figure 3-3.

Service Subroutines (Called by Realtime Programming System Assembler Language Programs Only)

The MFSL subroutines require an operating environment that includes the library work area and an interruption-handling facility. This environment is established by calls to the service subroutines. These subroutines are shown in Figure 3-3.

The FORTRAN IV user of MFSL need not be concerned with these subroutines because the compiler-generated object code calls the initialization and termination subroutines. The additional error exit can be invoked from an assembler subroutine operation in a FORTRAN IV environment.

The assembler user must call the initialization subroutines before calling any other MFSL functions, and should call the termination subroutine to release the resources acquired.

Subroutine category	Functions
Mathematical	Arc tangent, one or two arguments; cosine; divide doubleword integers; exponential functions; exponentiation; hyperbolic tangent; logarithms, common and natural; maximum value; minimum value; modular arithmetic; multiply doubleword integers; positive difference; sine; square root; transfer of sign.
EBCDIC conversion	EBCDIC to floating-point; EBCDIC to integer; floating-point to EBCDIC; integer to EBCDIC.
Error-checking	Function test; floating-point divide exception; floating-point overflow/underflow.
Service	Library work area initialization; library work area termination; abnormal termination routine specification. (The service subroutines are called by Realtime Programming System assembler language application programs only.)

Figure 3-3. MFSL noncommercial subroutines

Commercial Subroutines

The commercial subroutines included in Realtime Programming System MFSL Version 2 and Event Driven Executive MFSL aid in using FORTRAN IV, Series/1 assembler language, and Event Driven Language programs in commercial applications. These commercial subroutines perform output editing (formatting), data conversion, variable-length decimal arithmetic, and frequently needed utility functions. These subroutines are described in Figure 3-4.

Subroutine category	Functions
Output formatting	The EDIT subroutine prepares output in special formats. Typical uses of EDIT are to insert commas, supply leading blanks, float monetary symbols, and to display a CR (credit) symbol after negative numbers. EDIT is especially useful in preparing invoices, checks, and other commercial documents.
Data conversion	The A1DEC and DECA1 subroutines convert data between the FORTRAN IV A1 format (one character per 16-bit word, left-justified) and the D1 format (zoned decimal). The PACK and UNPAC subroutines convert data between the FORTRAN IV A2 format (two characters per 16-bit word) and A1 format.
Variable-length decimal arithmetic	The ADD, SUB, MPY, and DIV subroutines provide the basic arithmetic capability for decimal numbers with user-defined field lengths. The ICOMP/KCOMP comparison function subprogram and the NSIGN sign test subroutine provide further capability for decimal data handling.
Utility subroutines	The LCOMP and NCOMP function subprograms compare variable-length alphameric fields (A1 format) according to two different collating sequences: LCOMP uses the Series/1 collating sequence, and NCOMP uses the IBM 1130 collating sequence. The MOVE subroutine moves data from one area in storage to another. The FILL subroutine fills an area in storage with a specified value.

Figure 3-4. MFSL commercial subroutines

Chapter 4. Communication Products Support

IBM Series/1 communications product supports communications through the following:

- Realtime Programming System 4987 Programmable Communications Subsystem:
  - Preparation Facility
  - Execution Support
  - Extended Execution Support
- Additional communications support:
  - Remote Job Entry
  - Remote Management Utility

4987 Programmable Communications Subsystem Preparation Facility

The 4987 Programmable Communications Subsystem Preparation Facility (5719-CS0), hereafter referred to as the Preparation Facility, is a macro library that assists the programmer in writing function strings and in creating tables required by the 4987 Programmable Communications Subsystem to perform communication functions under the Realtime Programming System. A function string is a sequence of orders that defines a series of operations for the programmable communications subsystem. Orders are the instructions that enable the 4987 Programmable Communications Subsystem to perform useful communication functions.

The Preparation Facility consists of *order macro instructions* and *controller storage definition macro instructions*. These macro instructions allow the programmer to write source code for the controller storage image program. This code is then assembled by the Program Preparation Subsystem or the Base Program Preparation Facility.

Order Macro Instructions

Order macro instructions are a tool for programming the controller, which is the programmable portion of the programmable communication subsystem used to support a variety of communication protocols and devices. There is one order macro for each programmable communications subsystem order. The programmer can use these order macro instructions as source code for customized communication programs (function strings).

Controller Storage Definition Macro Instructions

Controller storage definition macro instructions allow the programmer to select function strings and to create the tables used by the controller to perform the communication functions required.

Source Code for Controller Storage Image Program

The source code for a controller storage image program consists of function string source code, controller storage definition macro instructions, and other control statements required by the program preparation system being used. When this source code is assembled and link edited, it produces a loadable controller storage image program containing function strings, tables, and data for use by the controller.

Controller Storage Image Program

A controller storage image program contains the function strings, tables, and data required by the controller to perform communication tasks. It exists as a loadable module resulting from the assembly and link-edit of the controller storage image program source code. The controller storage image program must be loaded into the controller. For additional information, refer to 4987 Programmable Communications Subsystem Execution Support.

Figure 4-1 provides an overview of the Preparation Facility and its relationship to the 4987 Programmable Communications Subsystem.

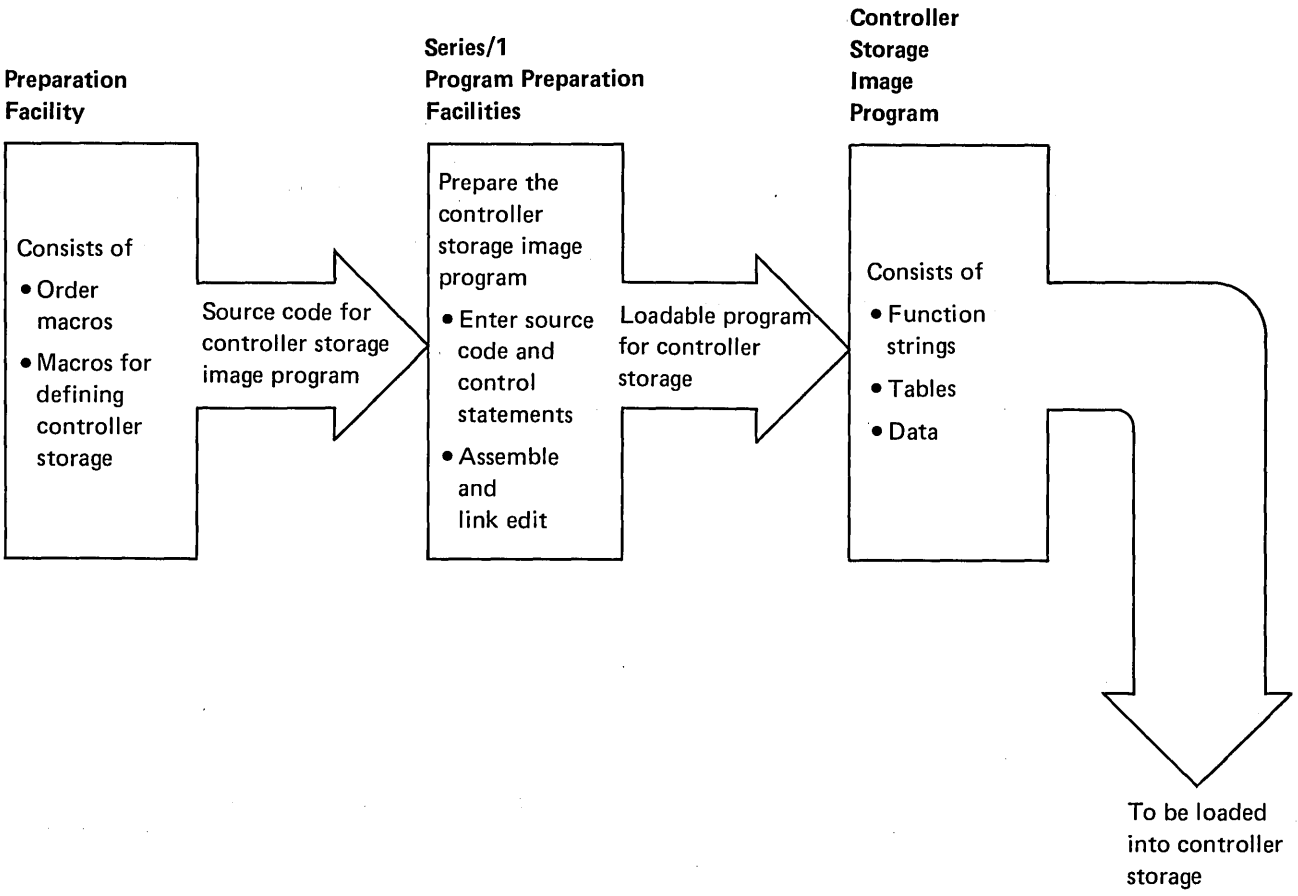


Figure 4-1. Preparation facility overview

4987 Programmable Communications Subsystem Execution Support

The 4987 Programmable Communications Subsystem Execution Support (5719-CS1), hereafter referred to as the Execution Support, provides an interface to the 4987 Programmable Communications Subsystem through the Realtime Programming System Version 2, Version 3, or Version 4. The Execution Support consists of the following:

- Utility
- Execution-time support
- Execution support macro instructions

Figure 4-2 illustrates the functions performed.

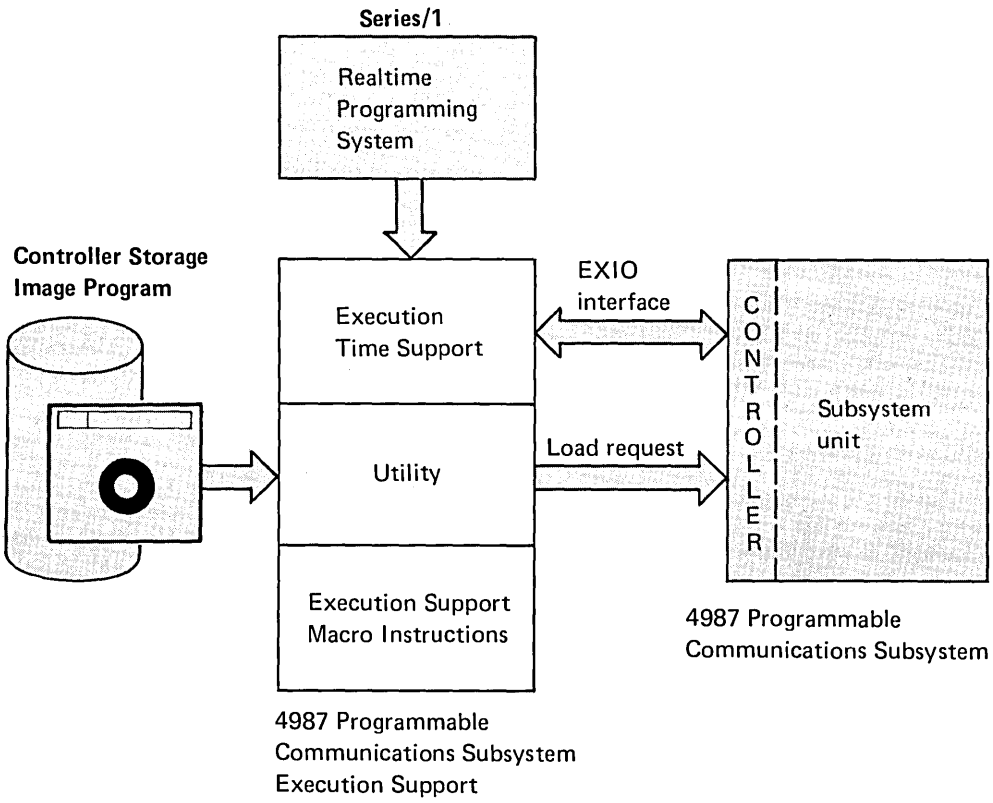


Figure 4-2. Overview of execution support

Utility

The utility allows the user to load a controller storage image program from disk or diskette to controller storage. The utility also allows the user to dump a line control block (LCB) from the 4987 to system operator station and the system message data set. The user can initiate the utility from the system console or from his application program. For a load request, the utility executes internal diagnostic programs before loading the controller storage.

Execution-Time Support

The execution-time support provides the user with an interface to the 4987 Programmable Communications Subsystem through EXIO communication support in the Realtime Programming System Version 2, Version 3, or Version 4. This support provides macro instructions, which enhance existing Realtime Programming System functions.

Through Realtime Programming System Version 2, Version 3, or Version 4, the following are provided:

- Issues machine I/O instructions to the 4987 Programmable Communications Subsystem
- Handles interrupts from the Programmable Communications Subsystem
- Returns errors and completion codes, except for controller end
- Interface to the Realtime Programming System Version 2, Version 3, or Version 4 trace facility for tracing channel activity

Execution Support Macro Instructions

The execution support macro instructions provide unique control blocks and tables required by some Programmable Communications Subsystem hardware commands.

4987 Programmable Communications Subsystem Extended Execution Support

The 4987 Programmable Communications Subsystem Extended Execution Support (5719-CS2) enables Realtime Programming System Version 4 applications to communicate through the 4987 Programmable Communications Subsystem (referred to as the 4987) to any supported processor or terminal in either point-to-point (switched and nonswitched) or multipoint networks, depending upon the capability of the individual terminals.

In multipoint networks, a Series/1 can be either the controlling or tributary station, or both.

The Extended Execution Support consists of the following:

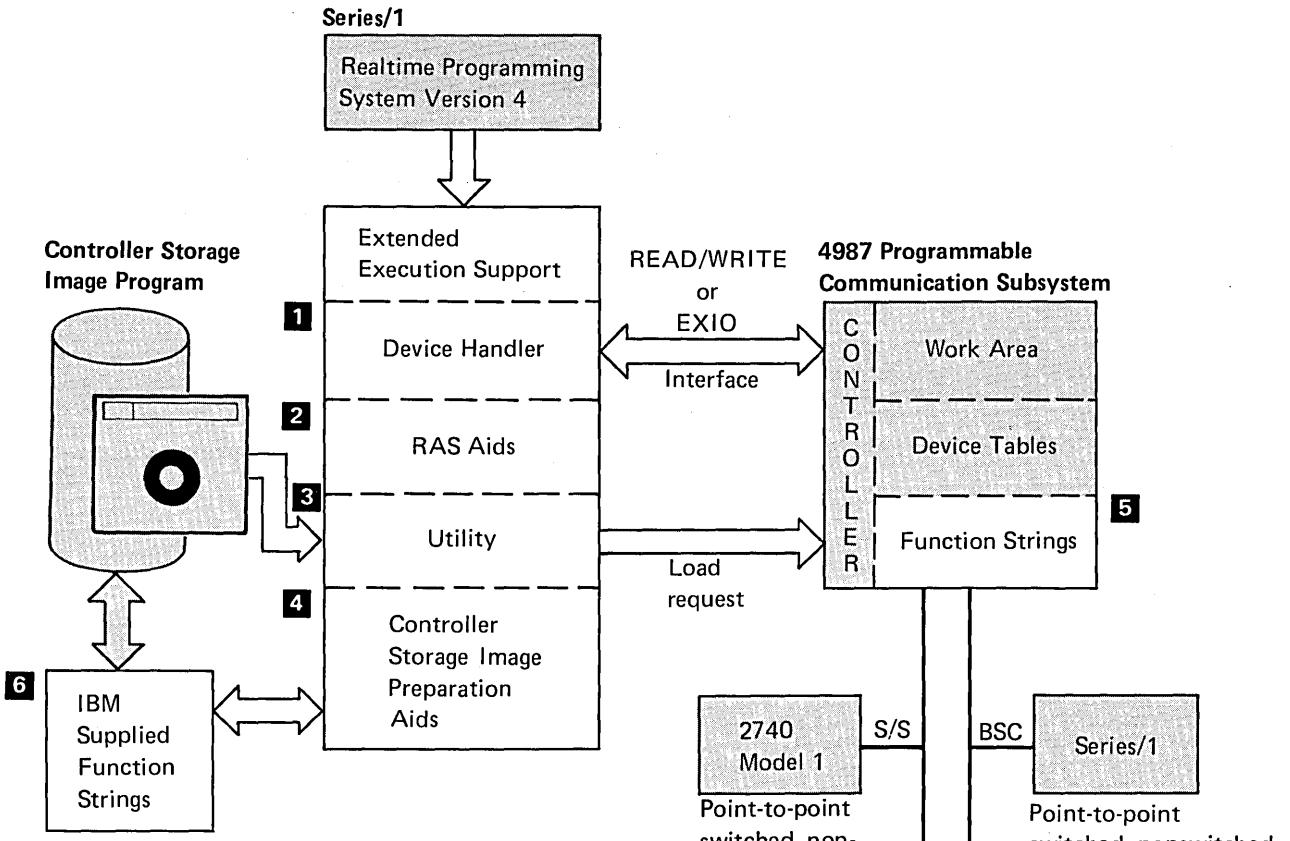
- Specialized device handler
- IBM-supplied 4987 programs (called function strings) and control character tables
- Reliability, Availability, Serviceability (RAS) aids
- Controller storage image preparation aids
- Loader utility

Figure 4-3 illustrates the functions performed.

Device Management

The Extended Execution Support specialized device handler provides the following:

- READ/WRITE macro level interface to the 4987 for supported terminals.
- An EXIO macro level interface to the 4987 for supported or non-supported terminals.
- Opening of lines on the 4987 for either READ/WRITE or EXIO access level regardless of the access level for which other lines have been opened.
- Configuration flexibility—Configuration description occurs at your program execution time. There is no requirement to define terminals or line or device characteristics during system generation.
- Association of a terminal with its 4987 program support through the controller storage image descriptor. The external definition of the controller storage image provides for transportability of your application programs to a variety of 4987 configurations.
- Error handling for 4987 errors.
- Dynamic allocation of data management control block storage—device related control blocks are allocated at start device time and line related control blocks are allocated at OPEN time, not at generation time.
- Online test facilities for the supported binary synchronous communications (BSC) and start/stop (S/S) devices.



1 Provides READ/WRITE macro level user interface for supported terminals, as well as EXIO macro level user interface for supported or nonsupported terminals.

2 Composite of Realtime Programming System RAS functions and RAS functions unique to the Extended Execution Support.

3 Loads the controller storage image program into controller storage (can be initiated from operator station or application program).

4 Consists of macros that can alter a default number of retry attempts, and define/create a controller storage image descriptor, which describes the elements of the controller storage image to the device handler.

5 Defines operations to be performed to accomplish a specific communications function. Listed below are BSC and S/S devices supported.

6 IBM supplied function strings for supported terminals.

Figure 4-3. Overview of extended execution support

IBM-Supplied Function Strings

The IBM-supplied 4987 programs, called function strings, are a sequence of orders that define operations to be performed to accomplish a specific communications function. You can:

- Select only the function strings to support those terminals in your communications network.
- Write your own function strings and combine them with IBM-supplied function strings. If you write your own function strings, you may want to order the IBM Series/1 4987 Programmable Communications Subsystem Preparation Facility (5719-CS0) to aid in preparing the controller storage image. User function strings can only be used with the EXIO macro level interface.

The IBM-supplied function strings handle the required line protocols; control the transmission of data communications to and from your buffer areas in the Series/1; and where possible, attempt transmission retries when line transmission errors are detected. The IBM-supplied function strings are provided in source and object format to support specific binary synchronous communications (BSC) devices and start/stop (S/S) devices.

In addition, a 4987 controller storage image (CSI) containing all of the IBM-supplied function strings and a corresponding controller storage image descriptor (CSID) are provided in both source and object format. They can be installed immediately to assist in developing applications for the IBM supported BSC and S/S devices.

Reliability, Availability, Serviceability (RAS) Functions

RAS functions in support of the Extended Execution Support are a composite of general Realtime Programming System RAS functions and Extended Extended Execution Support RAS functions. Among the RAS functions provided by the Realtime Programming System that support the Extended Execution Support are:

- Storage dump
- Direct access data dump
- Patch direct access data
- Patch storage
- Error message issuance

RAS functions provided by the Extended Execution support in conjunction with the Realtime Programming System are:

- Communications I/O trace
- Error retry, when not handled by the function strings
- Statistical data collection
- Execution time error detection
- Online terminal test
- Error logging facility

The Extended Execution Support RAS functions consist of a set of RAS operator commands to assist you in using the 4987 controller trace and dump facilities. The Extended Execution Support RAS operator commands provide a set of interactive commands that can be used to cause execution of the following RAS functions:

- Starting the RAS function
- Starting or stopping data and/or order tracing in the 4987
- Clearing the 4987 trace buffer
- Dumping portions or all of the 4987 controller storage to your disk/diskette data set, printer, or operator station
- Dumping a 4987 line control block to your disk/diskette data set, printer, or operator station
- Dumping the 4987 controller trace area to your disk/diskette data set, printer, or operator station
- Formatting and printing a 4987 line control block dump
- Formatting and printing a 4987 controller storage dump
- Formatting and printing a 4987 trace buffer dump
- Terminating the RAS function

Controller Storage Image Preparation Aids

Controller storage image preparation is accomplished by assembling and application building controller storage image program and its associate controller storage image descriptor. A controller storage image descriptor describes the elements of the controller storage image to the device handler. The controller storage image preparation aids consist of macros that allow the user to:

- Alter a default number of retry attempts by the 4987 for BSC and S/S devices
- Alter a default number of rings required by the 4987 after the first ring by an auto answer device
- Define and create a controller storage image descriptor

Loader Utility

The loader utility loads the controller storage image into controller storage. The loader utility can be initiated from the operator station using an operator command or from an application program. The loader utility performs the following functions:

- Executes the 4987 internal diagnostics prior to loading
- Reads the controller storage image from disk or diskette
- Loads the controller storage image into the 4987 controller storage
- Verifies that the load was accomplished without error

Devices Supported

The following BSC and S/S devices are supported:

- Another Series/1
- 3275 Display Station (Models 1 and 2)
- 3271 Control Unit (Models 1 and 2) with 3277, 3284, and 3286
- 2740 Communications Terminal (Models 1 and 2)
- 2741 Communications Terminal
- Teletype ASR 33/35 Terminals

Additional Communications Support

The following Programming RPQs provide additional communications support.

Remote Job Entry Programming RPQ

Remote Job Entry (RJE) provides the Series/1 Realtime Programming System user the ability to transmit jobs and receive output from a host System/370 having OS/VS2 and JES/2 installed. Support is provided for binary synchronous communication point-to-point (switched or nonswitched). Input is from the operator station, disk, or diskette; output is to printer, disk, or diskette. RJE runs as a task in a 16K-byte partition under the Series/1 Realtime Programming System. The following devices are supported:

- 4962 Disk Storage Unit
- 4964 Diskette Storage Unit
- 4973 Line Printer
- 4974 Printer
- Binary Synchronous Single Line Control (#2074)
- Binary Synchronous Single Line Control/High Speed (#2075)
- Binary Synchronous Communications 8-Line Control (#2093)
- Binary Synchronous Communications 4-Line Control (#2094)
- 4979 Display Station
- Teletype model ASR 33/35 or an equivalent ASCII device that can be used as an operator station and is attached to the system through the Teletypewriter Adapter Feature #7850

Remote Management Utility Programming RPQ

The Remote Management Utility provides the capability for a Series/1 operating under the Realtime Programming System Versions 2, 3, or 4 to communicate with an application in a host system through BSC attachments.

The Remote Management Utility transfers files between a host and a remote Series/1, facilitates the operation of the remote Series/1 in a loosely-coupled distributed data processing system, and runs as a task set under the Realtime Programming System on the Series/1.

The Remote Management Utility interacts with the host program by receiving commands from the host and returning codes to the host to signal the results of command execution. At the end of the transmission of a data set in either direction, it returns a count of records received or transmitted, allowing the host application program to build a session log. The following devices are supported:

- 4962 Disk Storage Unit
- 4963 Disk Storage Unit
- 4964 Diskette Storage Unit
- 4966 Diskette Magazine Unit
- Binary Synchronous Communications Single Line Control (#2074)
- Binary Synchronous Communications Single Line Control/High Speed (#2075)

Chapter 5. Special Device Support

Special device support is provided through the following:

- Series/1—System/370 Channel Attach Program
- 4969 Magnetic Tape Subsystem Support
- Realtime Programming System Screen Formatter
- 5250 Information Display System Attachment Support
- Multiple Terminal Manager
- Additional Device Support
  - 4978 Display Station Support
  - Transient Activity Tool
  - Address Translator Transient Support

Series/1—System/370 Channel Attach Program

The IBM Series/1—System/370 Channel Attach Program (5719-CA1 for the Realtime Programming System versions 3 or 4,* and 5719-CX1 for the Event Driven Executive Version 3) provides the capability for a Series/1 application program to communicate with a System/370 application program across a System/370 channel through the Series/1—System/370 Channel Attachment Feature.

The System/370 application program can execute under:

- Operating System Virtual Storage 1 (OS/VS1) with basic telecommunications access method (BTAM)
- Operating System Virtual Storage 2 (OS/VS2) with Multiple Virtual Storage (MVS)
- Single Virtual Storage (SVS) with BTAM for an IBM 3272 Control Unit to transfer data between application programs executing on the Series/1 and System/370 systems
- DOS/VSE with BTAM-ES

Note: See Figure 5-1 for a pictorial description.

Channel Attach Program

The Channel Attach Program provides 32 data ports to the attached System/370. Each port can be used independently. The Channel Attach Program:

- Establishes, controls, and terminates access between Series/1 application programs and the channel attach device
- Manages input/output transfers between Series/1 programs and the channel attach device

* All references to the Realtime Programming System Version 3 or 4 in this section are for Version 3, Modification Level 1 and later modifications, and Version 4, May 1979 and later modifications.

- Communicates with the System/370 over 32 ports (device addresses) per channel attach device
- Performs error logging
- Traces Series/1 input/output commands, ending status interrupts, and attention interrupts from the channel attach device when "trace" is started by the Series/1 operator or an application program
- Handles interrupts from the channel attach device

Note: Due to timing constraints, Channel Attach Program trace records are not written to the operator station.

The Channel Attach Program basic level of access:

- Provides condition codes of operate input/output (OIO) and interrupts
- Posts the application program when an attention interrupt occurs and supplies the interrupt information byte (IIB) as well as the status of the channel attach device
- Manages data ports to avoid conflict among Series/1 users
- Performs error recovery and retry wherever possible

Channel Attachment

The Series/1—System/370 Channel Attachment connects a System/370 selector or block multiplexer channel to the input/output interface of a Series/1 processor. The channel attach device transfers data under joint consent between System/370 and Series/1 application programs. Up to eight attachments can be connected on any one System/370 input/output channel. The maximum number of attachments to the Series/1 input/output interface is determined by the addressing and physical limitations of the system. The channel attach device consists of:

- 4993 Series/1—System/370 Termination Enclosure
- Series/1—System/370 Channel Attachment Feature 1200

The minimum software and hardware requirements to support the Channel Attach program are:

- System/370 (Model 135–168) or a 3031, 3032, 3033, 4331, or 4341 processor with selector or block multiplexer channel to operate under:
  - OS/VS 1 BTAM, or
  - OS/VS 2 MVS or SVS BTAM
- Series/1 Processor with at least 64K bytes of main storage operating under the Realtime Programming System Version 3 or 4 or Event Driven Executive Version 3
- Series/1—System/370 Channel Attachment (described above)

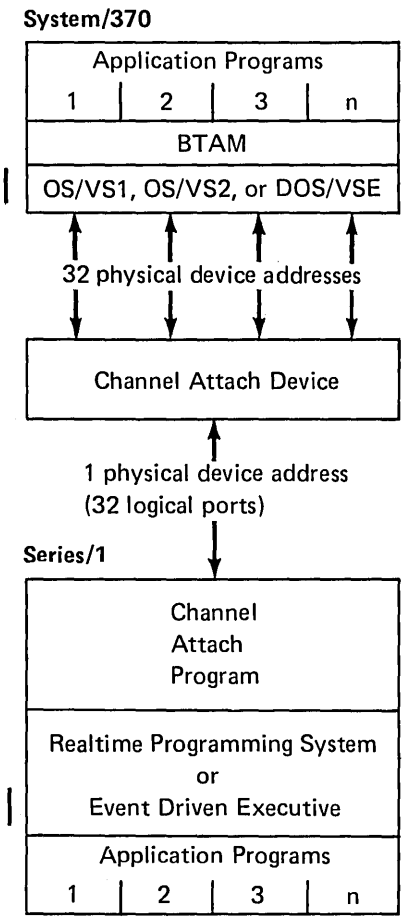


Figure 5-1. A Series/1—System/370 Channel Attach System

4969 Magnetic Tape Support

The IBM Series/1 4969 Magnetic Tape Subsystem Support (5719-TA4) provides the following for the 4969 Magnetic Tape Subsystem:

- I/O functions
- Label support
- Device control functions
- Tape utilities
- Tape exerciser

The IBM Series/1 4969 Magnetic Tape Subsystem Support runs under the Realtime Programming System Version 4 and supports the COBOL, PL/I Version 2, and Sort/Merge licensed programs. An overview of the 4969 Magnetic Tape Subsystem Support and its relationship to the 4969 Magnetic Tape Subsystem are shown in Figure 5-2.

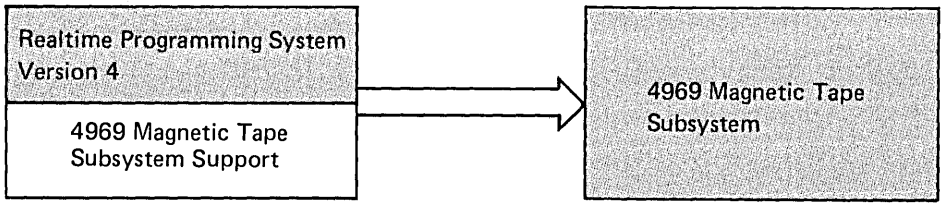


Figure 5-2. Overview of IBM Series/1 4969 Magnetic Tape Subsystem Support

I/O Functions

To read and write tapes, OPEN/CLOSE functions and three levels of access are provided:

- Logical (GET/PUT)
- Physical (READ/WRITE)
- Basic (EXIO)

Facilities for activating the 4969 Magnetic Tape Subsystem Support and for building the control blocks required to add tape I/O to a Version 4 system are also provided.

Retry procedures are used when data transfer errors occur; standard Realtime Programming System error logging and I/O tracing functions are used to accumulate error statistics and provide debugging aids.

Label Support

Tapes can either have IBM standard labels that are compatible with DOS/VS or be unlabeled. The labeled support provides for writing header and trailer labels for output processing and verifying header and trailer labels on input processing.

Device Control Functions

These functions provide capability to do the following:

- Forward space or backspace a record or file
- Rewind a tape
- Write a tape mark
- Use multivolume files through the volume-switching facilities provided

Tape Utilities

The following are part of the IBM Series/1 4969 Magnetic Tape Subsystem Support:

- The *backup* utility copies the contents of a disk, a logical volume, or a data set to or from tape.
- The *copy* utility copies data from one tape to another.
- The *initializer* writes a volume label and file label.
- The *DSD* utility builds a tape DSD into a DSDT on disk.

Realtime Programming System Screen Formatter

The Series/1 Realtime Programming System Version 4 Screen Formatter is made up by two separate licensed programs. They are:

- Screen Format Design Aid Utility (5719-SF1)
  - Interactive
  - Option-Driven
  - User defines display device attribute characters
  - User can design and test screen format prior to use by user program
  - User can obtain hard copy of screen formats for reference
- Presentation Support (5719-SF2)
  - Designed to improve user productivity
  - High level language interface for COBOL, PL/I Version 2, macro assembler, and FORTRAN IV

Both licensed programs are required on systems where screen formats are created, changed and saved in screen format libraries on disk or diskette storage. Presentation Support (5719-SF2) is designed to operate on production systems; it does not require the Screen Format Design Aid Utility.

The Screen Format Design Aid Utility executes in a 16K Byte partition. It is menu driven and interactive. The Utility is used to create, change, test and save screen formats in libraries on IBM 4962, 4963 Disk Storage or 4964, 4966 Diskette Units. This Utility requires the Presentation Support (5719-SF2) licensed program.

The Screen Format Presentation Support (5719-SF2) is used on production systems and operates as supervisor code and composite modules which are combined with user applications. An SVC interface is provided for high level languages (COBOL, PL/I, or FORTRAN IV). Presentation Support allows user written applications to use multiple display stations. Presentation Support provides the following functions (from a predefined and created Screen Format Library):

- PUT FORMAT (write screen format to display device)
- PUT STRUCTURE (write unprotected data on display device)
- GET STRUCTURE (read unprotected data from display device)

5250 Information Display System Attachment Support

The IBM Series/1 5250 Information Display System Attachment Support (5719-TA1), hereafter referred to as the 5250 Attachment Support, provides definition and execution time facilities to assist the user in the control of 5250 stations that are attached to the Series/1 by a 5250 Information Display System attachment.

The 5250 Attachment Support runs under the control of the Realtime Programming System Version 4 (5719-PC4) and provides the following support:

- Station attachment device support
- 5250 station verification test facility
- 5250 Information Display System attachment initialization
- Screen formatting assist macros
- Alternate system console support
- Utility functions
- Interface to PL/I Version 2, COBOL, and FORTRAN IV
- Print screen capability

Device Support

The device support facility enables the user's applications to communicate with the 5250 Information Display System attached to the Series/1. The device support provided is at a READ/WRITE macro level only.

Verification Test Facility

The verification test facility enables the user to verify operational status during installation, normal operation, and servicing of the 5250 stations attached to the Series/1. The following verification tests are provided:

- Display Verification
  - Display attributes
  - Displayable characters
  - Specified input fields
  - Function keys and features
- Printer Verification
- Configuration Data
- 5250 Error Log Information

5250 Information Display System Attachment Initialization

The 5250 Information Display System Attachment initialization facility loads the controller storage image into the 5250 Information Display System Attachment. The user can initiate this facility from the operator station or the initial program load (IPL) options data set using the Realtime Programming System start device operator command or from the user's application using the Realtime Programming System start device macro. The 5250 Information Display System Attachment initialization facility performs the following functions:

- Opens the data set referenced by the macro or by the command that starts the device
- Reads the controller storage image from disk or diskette
- Loads the controller storage image into the 5250 Information Display System attachment

Screen Formatting Assist Macros

The screen formatting assist macros help the user construct output data streams and screen formats for the 5250 display stations. Data streams are used to write data to a 5250 Information Display System and consist of data and control information with which the user can control the placement and attributes of the data.

Alternate System Console Support

The alternate system console support enables the user to use a 5250 display station as the alternate system console after installation and initial system generation. A 4979 Display Station or Teletype Model ASR 33/35 or equivalent ASCII device must be used as the system console during initial system generation and Realtime Programming System installation.

Utility Functions

The following functions are available via invocation of a utility task set:

- Build data set definitions (DSDs) for the 5250 stations attached to the 5250 Information Display System Attachment feature
- Change existing 5250 DSDs

Operator commands are provided to allow the user to:

- Stop a 5250 Information Display System Attachment feature and free the storage that was required to service the 5250 Information Display System Attachment feature
- Reset dates and counters in the error log data set for the 5250 stations

Multiple Terminal Manager

The Multiple Terminal Manager (5799-TCY for the Realtime Programming System, 5719-MS1 for the Event Driven Executive Versions 1 and 2, or 5719-MS2 for Event Driven Executive Version 3) is a library of functions to aid the user in creating a transaction-oriented multiple terminal application. The user application will be loaded and run in the same partition as the Multiple Terminal Manager.

This application uses the Realtime Programming System or Event Driven Executive to provide support for local 4978 and 4979 Display Stations in formatted full-screen mode and teletypewriter terminals attached via the asynchronous communications adapter in unformatted mode.

Interactive or transaction oriented applications such as; inquiry, file update, data collection, and order entry can be implemented using the following access methods:

- Indexed access method
- Direct access method
- User provided access routines (sequential, direct, or other)

Any direct access device supported by the appropriate access methods can be used with the Multiple Terminal Manager. Terminal, file, and program management services are also provided. Terminal I/O operations are performed automatically. File I/O operations are performed at the request of user-written application programs. An 8-character program name or a function/predefined key is used to determine which program will process the transaction. The following devices are supported:

- 4962 Disk Storage Unit
- 4963 Disk Storage Unit
- 4964 Diskette Unit
- 4966 Diskette Magazine Unit
- 4978 Display Station
- 4979 Display Station
- 3101 Display Terminal (Models 10, 12, and 13) attached with an asynchronous communications adapter
- 3101 Display Terminal (Models 20, 22, and 23) with an asynchronous communications adapter (Event Driven Executive only)
- 5250 Information Display System (Realtime Programming System only)
- 4973 Line Printer
- 4974 Printer

The Multiple Terminal Manager supports the following Series/1 languages:

- FORTRAN IV
- COBOL
- PL/I Version 2
- Assembler
- Event Driven Language (EDL)

Additional Device Support

The following Programming RPQs provide additional device support.

4978 Display Station Support Programming RPQ

The 4978 Display Station Support extends the Series/1 Realtime Programming System to provide read/write level support for the 4978 Display Attachment and to permit the 4978 to be substituted for the 4979 as the system console device (except in the starter system delivered from IBM). The functions provided by this support are:

- Scroll
- Write full/part screen
- Read full/part screen
- Erase full/part screen
- Multiple interrupt keys
- Read modify
- Character font definition functions
- Attachment initialization
- Attachment control store
- Tone alarm

The following devices are supported:

- 4962 Disk Storage Unit
- 4964 Diskette Storage Unit
- 4978 Display Station RPQ

Transient Activity Tool Programming RPQ

The Realtime Programming System Transient Activity Tool Programming RPQ provides system transient I/O frequency counts by module as an aid in Realtime Programming System Version 3 performance tuning.

Command action routines provide for turning transient monitoring on and off by an operator command. The command action routines also provide the ability to dynamically load and unload system transients specified by the user.

Transient monitor routines capture and record information concerning system transients during execution of a user application. Transient monitor routines also capture I/O frequency counts.

The report generator prints a sorted formatted listing of module names, sizes, relative disk addresses, load, fetch, refresh, and total counts for modules with non-zero total counts. The following devices are supported:

- 4962 Disk Storage Unit
- 4973 Line Printer
- 4974 Printer
- Teletype Model ASR 33/35 or an equivalent ASCII device

Address Translator Transient Support (ATTS) Programming RPQ

ATTS provides the Realtime Programming System Versions 1, 2, and 3 user with a system-controlled pool of transients located in secondary storage (outside the first 64K bytes of physical storage). This support allows the user to generate a heavily transient system and still have high performance. The 4955 Processor Models B and D with a Storage Address Relocation Translator and the 4955 Processor Model E are supported. (This capability is implemented within Version 4 under the DTPM feature.)

Chapter 6. Data Management and Access Support

Series/1 data management and access is supported through the following:

- Sort/Merge
- Indexed Access Method
- Additional support:
  - Indexed Access Method
  - Basic Sort
  - Disk Spooling
  - Data Collection Interactive Programming RPQ

Sort/Merge

IBM Series/1 Sort/Merge (5719-SM1 for Realtime Programming System, 5719-SM2 for Event Driven Executive) handles the sorting and merging of records from up to eight input data sets into one output data set in either ascending or descending order. The user specifies one or more control fields in the records to be sorted; the program then compares the control fields to determine the relative sequence of the records.

IBM Series/1 Sort/Merge requires a Series/1 processor with a minimum of 48K bytes of main storage operating under control of the Realtime Programming System Version 3 or Version 4 or the Event Driven Executive Basic Supervisor and Emulator. Through the Realtime Programming System it can be initialized with an EXEC statement using the Job Stream Processor or with a system macro instruction from a user routine written in Series/1 assembler language. Through the Event Driven Executive it can be initialized with a $L operator command, invoked through the LOAD instruction from a user routine written in Event Driven Language, or the $JOBUTIL utility.

Sort/Merge provides the following:

- Address sort
- Record sort
- Record summary sort
- Merge

Address Sort

Address sort produces an output data set of four-byte relative-record numbers, one for each record in the input data set. A user program can then use the record numbers to access the records in the input data set in the ascending or descending sequence specified for the sort.

Record Sort

Record sort produces an output data set of records sorted in either ascending or descending order. The user determines the format and content of the sorted output record. The output record can be comprised of either the entire input record or one or more fields in the input record.

Record Summary Sort

Record summary sort accumulates totals from one or more user-specified fields (called summary fields) in the input records and provides one summary record for each set of input records having control fields with the same value (for example, in an inventory file, records with the same part number).

The summary record contains the totals for each summary field specified; in addition, it can contain the same user-specified data as allowed for record sort.

Merge

Merge combines up to eight previously sorted data sets, collating them in their previous ascending or descending sequence into one consecutive output data set.

The merged record consists of the entire input record and, at the option of the user, the control fields specified for the merge.

Magnetic Tape Support

IBM Series/1 4969 Magnetic Tape Support, licensed program (5719-TA4) can be used for SORTIN and SORTOUT.

Indexed Access Method

The IBM Series/1 Indexed Access Method (5719-AM1 for Realtime Programming System, 5719-AM3 for Event Driven Executive) is a data management licensed program that operates under the IBM Series/1 Realtime Programming System Versions 3 or 4, or the Event Driven Executive. The Indexed Access Method provides interfaces to build and maintain an indexed data set and to access, by key, the user records in the data sets. In an indexed data set, each user record is identified by the contents of a predefined field called a key. The Indexed Access Method builds into the data set an index of keys that provides fast access to the user records.

In the Realtime Programming System, the Indexed Access Method consists of a set of macros and modules. The modules are included in the Realtime Programming System at system generation time in the same way as a user-written supervisor program is included. The macros generate a supervisor call (SVC) interface to the Indexed Access Method task from the user task. In addition, a utility program is provided to create Indexed Access Method files. In the Event Driven Executive, the Indexed Access Method is a separately loaded program that is invoked via a CALL interface from the user task.

The Indexed Access Method offers the following features:

- Efficient processing of direct and sequential files. Multiple levels of indexing are used for direct access. Sequence chaining of data blocks is used for sequential access.
- Support for high insert/delete activity. Inserts can be made in place because of free space distribution throughout the data set and at the end. Space provided by deletions can be immediately reclaimed.
- A single data set can be accessed by several requests concurrently. These requests execute from the same or different programs, tasks, or task sets. Data integrity is maintained by a block and record level locking system that prevents access to an index or data record while that record is being modified.
- Implementation as a task—a single copy of the Indexed Access Method coordinates all requests. Queues, control blocks, and a buffer pool reside in system storage. The buffer pool in the system task supports all requests and optimizes the space required for physical I/O; in the user task, the only buffer required is the one for the record currently being processed.
- Immediate Write option provides the capability of specifying that all modified blocks are to be immediately written back to the file.
- User control of the Write/Verify option allows the user to specify whether or not the hardware Write/Verify option is to be invoked on file writes. (Realtime Programming System only).
- Extract data from file control block option provides the capability of extracting information about a file such as key length, key displacement, block size, record size, and more detailed information regarding the file structure.

Routines using the Indexed Access Method interfaces are written in assembler (Event Driven Language for the Event Driven Executive) and can be included in programs written in any language that supports the calling of assembler language or Event Driven Language routines. The interfaces provide the capability to:

- Define and create an indexed data set
- Open and close the data set
- Load sequenced records
- Retrieve records sequentially or directly by key
- Insert, delete, and update records
- Obtain information about the data set

Additional Support

The following Programming RPQs provide additional data management and access support.

Indexed Access Method Programming RPQ

The Indexed Access Method provides keyed access to user data to support a variety of applications ranging from batch processing to multi-user interactive applications. It is supported by the Realtime Programming System Versions 1 and 2. It supports:

- Files that have high add/delete activity (such as open order files) to minimize performance degradation
- Multiple tasks sharing the same data files

PL/I Version 1 and FORTRAN users can write macro assembler language subroutines to be called by application programs written in those languages. The following devices are supported by the Programming RPQ:

- 4962 Disk Storage Unit
- 4964 Diskette Storage Unit

Basic Sort Programming RPQ

Basic Sort provides a set of functional modules to be used in conjunction with the Series/1 Realtime Programming System for sorting disk or diskette-based data sets into ascending or descending sequence. IBM provides a macro source module and object modules for the user to assemble into a program. The call interface to the object modules is generated by the expansion of the Basic Sort macro with user-defined parameters. The following devices are supported by this Programming RPQ:

- 4962 Disk Storage Unit
- 4964 Diskette Storage Unit
- 4973 Line Printer
- 4974 Printer

Disk Spooling Programming RPQ

Disk Spooling provides Series/1 Realtime Programming System users with a method of sequentially buffering and retrieving variable-length text records, associated with multiple reports, on disk. When spooling is active, print records (133K bytes maximum) are written to a user-allocated spool data set. This support is available to programs using the Series/1 Realtime Programming System's write or put level access. This includes programs written in assembler language, PL/I, or FORTRAN. The following devices are supported:

- 4962 Disk Storage Unit
- 4973 Line Printer
- 4974 Printer

Data Collection Interactive Programming RPQ

The Data Collection Interactive Programming RPQ provides programmable interface between the 523X Entry Station Direct Attach device and a Series/1 processor operating under the Event Driven Executive operation system. It allows for central data collection from 5234 Time Entry Stations, 5235 and 5236 Data Entry stations, and 5239 Value Read Modules.

This support consists of:

- System Personalization Functions—system personalization functions allow the user to define (via prompts at a 4978/4979 terminal) actions to be undertaken at the entry stations. Personalization once established is transportable and modifiable.
- Routing/Formatting Routines—the routing/formatting routines interface with the input/output control system (IOCS) to handle incoming data (in the form of 180-byte records) from the Series/1 data entry loop master scheduler attachment.
- IOCS Hardware Interface—IOCS provides the interface to the hardware required for controlling the 523X data entry loops. Each data entry loop (controlled by a multiplexer) provides the capability of collecting data from up to four entry stations (any combination).

Note: The IBM Series/1 Data Collection Interactive Programming RPQ, P82600 (5799-TDE) will be available to operate in the Event Driven Executive Version 3 environment in January, 1981.

Appendix A. Control Program Support

Control Program Support

Control Program Support Programming RPQ (5799-TAA) is a set of object modules which provide supervisory functions and I/O support which can be link edited with user code to create a nucleus and application program. Control Program Support is not provided as an integrated operating system. It performs functions which can be used with user developed code to produce a particular supervisor for a given application. The user is responsible for controlling the machine environment.

System Overview

The Control Program Support Supervisor Functions provide the user with functions for controlling program execution and overlapping the input/output operations with program execution. A disk or a diskette is required to load an application. Additional partial storage loads may be loaded from disk or diskette to main storage. The operator station is not required at IPL time, but it is supported if present.

The Control Program Support and its extensions provide facilities for: task and data management support, device support, and service support.

Task and Data Management Support

The Control Program Support extensions that provide facilities for task and data management support are:

- Extensions I Programming RPQ
- Extensions II Programming RPQ
- Extended Function Programming RPQ
- Address Translator Support Programming RPQ
- Index Access Method Programming RPQ
- Commercial Arithmetic Programming RPQ

Extensions I

Control Program Support Extensions I Programming RPQ (5799-TAL) provides a set of functional modules to be used in conjunction with Control Program Support for I/O Queuing, Data File Integrity, and Buffer Pooling.

I/O Queuing. Provides a facility to queue I/O requests to a device, even if the device is currently busy or if other user requests to that device are pending. Control is returned to the user at the next sequential instruction following an asynchronous return to some address upon completion of the I/O.

Data File Integrity. A file protection feature is provided by means of a "Lock" option on a disk or diskette read request. I/O requests subsequent to the locking of a data file will be queued to the device's lock chain until the data set is unlocked and the I/O request queued to the device.

Buffer Pooling. Provides dynamic storage allocation for temporary storage requirements. Each pool will be segmented into fixed blocks (bytes). The block size and number of pools are defined by the user.

Immediate control will be returned to the user with either the address of the required buffer or an indication that the pool does not contain an available buffer of the requested size. Actual buffer acquired may be larger in size than explicit value requested.

Extensions II

Control Program Support Extensions II Programming RPQ (5799-TAQ) provides all the facilities of Extensions I (5799-TAL) plus an additional set of functional modules to be used in conjunction with Control Program Support for Data Editing, EBCDIC/Binary Conversion, and Time/Data Reference.

Data Editing. The Edit capability permits the application programmer to perform commercial editing of EBCDIC character strings. The functions provided are:

- Suppress non-significant zeros, insert dollar signs, commas, decimal points and slashes, insert minus sign or credit symbol, and specify where suppression of leading zeros should stop by using the EDIT macro.
- Test input data against an edit word(s) and optionally, compress the input data by inserting leading zeros while deleting dollar signs, commas, decimal points, and slashes while maintaining all signs of fields (positive or negative) by using the EDIT macro.

EBCDIC/Binary Conversion. This routine converts a single or doubleword binary value to a decimal EBCDIC character string or converts a decimal EBCDIC character string to a single or doubleword binary value. It inserts a blank or minus sign into the high-order character of EBCDIC field and signs the binary value accordingly. The range is a signed number between + or − 2,147,483,648.

Time/Date Reference. This routine will initialize and support a time-of-day clock and calendar.

Task Scheduling. This routine allows scheduled activation and deactivation of selected tasks by time of day in conjunction with the cyclic calendar support provided in this Programming RPQ. Provisions are made for 24-hour rollover and leap year checking. Optionally, Scheduler Control Block Management can be accomplished under program control. This allows dynamic modification of the scheduler data to enable, disable, or delete a task.

Extended Function

The Control Program Support Extended Function Programming RPQ (5799-TBQ) provides a set of functional modules that provides extended function for disk and printer devices and enhances system usability.

Hardware Support. The Control Program Support Extended Function Programming RPQ provides support for the following hardware units:

- *4973 Line Printer Models 1 and 2.* The 4973 Line Printer Models 1 and 2 are supported at a level equivalent and compatible to the 4974 Printer support. Support for concurrent operation of multiple 4973 and 4974 printers is provided with identical user interfaces. The 4974 Printer support and the 4973 Line Printer support provides line spacing of 6 or 8 lines per inch.
- *4962 Disk Storage Unit Models 3 and 4.* The Disk File Support is extended to support one or more 4962 Disk Storage Units Model 1, 1F, 2, 2F, 3 or 4 in any combination. The user's interface to all disk and diskette devices is identical.
- *4962 Disk Storage Unit Models 1F and 2F.* Support of the 4962 Disk Storage Unit is extended to include the fixed head facility of that unit.

Software Support. The Control Program Support Extended Function includes the following software enhancements:

- *Table of Contents.* Provides the ability to create, maintain, and dynamically locate symbolic files on all models of the 4962 Disk Storage Unit and 128 byte sector formatting on the 4964 Diskette Unit. Additionally, there is transportability between diskettes written on a system running Control Program Support and a Realtime Programming System configuration. The utilities may run standalone or online in a system overlay area.
- *Operator Console Support.* This function provides the user the capability to send messages to 4978, 4979, or Teletype devices operating as operator consoles. Messages can be informational only or informational with a request for a reply or acknowledgement. Messages requiring a reply are numbered for operator convenience. Two groups of macros are provided. The first group provides for an interface to the I/O communications support routines. The second group is used to construct the messages and control blocks.
- *Relocating Loader/Overlay Manager.* This function provides an easy method of invoking program and subprogram overlays that are disk or diskette resident in a symbolic directorized file and to have the system manage the overlay areas.

Address Translator Support

Address Translator Support Programming RPQ (5799-TBT) provides programming support for the Series/1 Storage Address Relocation Translator under the Control Program Support Programming RPQ. It provides the following features:

- Allows addressing of physical storage above 64K bytes to the upper limit of the processor.
- Supports up to eight user address spaces
- Provides for common mapping of storage loads across address spaces
- Provides for multiple tasks and storage loads per address space
- Provides for the user of the Overlay Management/Relocating Loader (Extended Function Programming RPQ) within each user address space
- Provides a debugging package which supports a Storage Address Relocation Translator environment
- Allows grouping of address spaces by classes
- Provides for the queuing of programs by address space class
- Provides for queuing of programs by name (Symbolic program file support).

Indexed Access Method Support

Index Access Method Programming RPQ (5799-TAH) extends Control Program Support to provide facilities for building, maintaining, and retrieving data files on 4962 disk or 4964 diskette. Each index entry contains an identifying field (key) and the disk address of the specific data record. An index of the keys provides fast access to any data record. It includes the following features:

- One Read/Write statement gets both index and data
- Index can be referenced and/or updated independently or concurrently with data file operations
- Data file can be referenced independently
- User may build one or more alternate indexes for a data file
- Deleted records may be replaced or flagged
- Records can be added to the end of data file
- Reorganization function provided for index
- All errors are logged to either a 4962 disk or 4964 diskette

Commercial Arithmetic

Commercial Arithmetic Programming RPQ (5799-TBD) provides a set of functional modules to be used in conjunction with Control Program Support for performing packed decimal arithmetic. The Commercial Arithmetic routines consist of a set of macro source modules and object modules distributed on diskette. The ability to add, subtract, multiply, divide, and compare up to fifteen packed decimal digits per operand is provided. A separately callable routine provides conversion between EBCDIC (unpacked) and packed decimal formats.

These routines provide for accurate results to fifteen decimal digits. If an insufficient number of digits is assigned for the result, the answer will be truncated on both sides of the designated decimal point.

Device Support

The Control Program Support extensions that provide facilities for device support are:

- Binary Synchronous Communications Support Programming RPQ
- 4963 Disk Support Programming RPQ
- 4979 Display Station Support Programming RPQ
- 4978/4979 Display Station Support Programming RPQ
- Disk Table of Contents Support Programming RPQ
- Sort/Merge Support Programming RPQ
- Disk Spooling Support Programming RPQ
- Format/Print Support Programming RPQ
- Autocall Support Programming RPQ
- Magnetic Stripe Card Reader Programming RPQ

Binary Synchronous Communications Support

Binary Synchronous Communications Control Program Support (5799-TAF) is an extension to Control Program Support to provide Read/Write support for the Series/1 binary synchronous communications features. It includes the following features:

- Point-to-point communications
- Read/Write level with transparency
- Auto answer
- Trace facilities
- Errors logged to 4962 disk or 4964 diskette
- The Series/1 system appears as an IBM System/3 in a communications environment (point-to-point only)

4963 Disk Support

The 4963 Disk Support Programming RPQ (5799-TCZ) provides the following support for the 4963 Disk Storage Unit:

- Connect function to connect any disk or diskette and return device extents.
- Read and Write function support.
- Scan function support to scan a maximum of 256 sectors for up to a 256 byte search argument.
- Supports 4962 and 4963 Disk Storage Unit, and 4964 Diskette (all models).

4979 Display Station Support

The 4979 Display Station Programming RPQ (5799-TAE) CRT screen handler facility defines a screen image for the 4979 Display Station. This facility offers the following functions:

- Write Full/Part Screen
- Scatter Write
- Read Full/Part Buffer
- Interface to user routines
- Screen Formatting
- User-provided DCB for fixed format

4978/4979 Display Station Support

The 4978/4979 Display Station Programming RPQ (5799-TAK) CRT Screen Handler facility defines a screen image for the 4978 Display Station. This facility offers the following functions:

- Scroll
- Write Full/Part Screen
- Read Full/Part Screen
- Erase Full/Part Screen
- Shift Followed by Erase
- Interface to user routines
- Screen formatting
- Multiple Function Keys
- Read Modify Function
- Adapter Storage Initializer
- Error Recording and Recovery
- Adapter Storage Loads

4963/4966 Save/Restore

Control Program Support 4963/4966 Save/Restore Programming RPQ (5799-TDK) provides a set of functions contained in a load module which is IPLable from a 4964 Diskette Unit. These functions together provide the ability to save all or any part of the data stored on a 4963 Disk Subsystem and to store the saved data on a 4966 Diskette Magazine Unit; subsequently, this data can be restored to its original location on the same 4963. It includes the following three functional elements:

- Stand-alone Monitor
- SAVE Utility
- RESTORE Utility

The use of this Programming RPQ requires the following minimum Series/1 hardware:

- 4952, 4953, or 4955 Processor with 32K bytes of storage
- 4963 Disk Subsystem Attachment Feature (3590)
- 4966 Diskette Magazine Unit Attachment Feature (1205)
- 4964 Diskette Unit Attachment Feature (3581)
- 4963 Disk Storage Unit Subsystem, Model 58A, 58B, 64A, or 64B
- 4964 Diskette Unit
- 4966 Diskette Magazine Unit

Disk Table of Contents Support

The Disk Table of Contents Programming RPQ (5799-TAW) provides program modules to create, maintain, and dynamically locate symbolic files on a 4962 Disk Storage unit. Up to 256 files are allowed; each file can contain up to 256 members.

The support consists of a dynamic locate function and a utility program providing four functions: Format, Update, List, and Pack. The utility is provided as both a standalone storage load and a set of object modules which can be link edited into the user's application program and invoked by the user's application program.

Sort/Merge Support

Sort/Merge Programming RPQ (5799-TAT) provides a set of functional modules to be used in conjunction with Control Program Support for sorting disk or diskette based data files into ascending or descending sequence.

The Sort/Merge routine is macro based and is assembled into a program by the user. A macro source module and object modules, all on a diskette, will be provided. The call interface to the object modules will be generated by the expansion of the Sort/Merge macro with user-defined parameters.

Input. The input to this expanded macro routine is the disk or diskette file to be sorted. Multiple input data sets are supported, up to a total of seven. Input record length must be defined by the user. Maximum record length is 1024 bytes. The maximum sum of key lengths is 248 bytes, and up to 12 sort fields can be specified. Control fields (keys) can be in different locations in the records. They can be sorted in ascending or descending sequence, or mixed (some ascending and some descending).

Output. This program creates a tag file as output. It is the user's responsibility to provide the routine which uses the tag file to construct an output file consisting of the following two segments:

- The first four sectors will contain a doubleword binary count of the number of tag words in the output file and a copy of the input/output IOCBs.
- The remainder of the file will consist of records containing a doubleword binary tag and the data keys used to perform the sort function. (*Note:* The data key(s) appended to the doubleword binary tag yield a user defined record size.)

Each entry in the output file consists of two binary words (doubleword), whose value will be in the range of +1 through *N*, followed by the data keys associated with that input record. Each binary entry of the tag file represents the number of a record in the input file. The tag numbers appear in the tag file in the sequence dictated by the user provided sort parameters.

Disk Spooling Support

Disk Spooling Programming RPQ (5799-TAY) provides a method of sequentially buffering and retrieving variable length text records, associated with multiple reports, on disk. Text records of variable length (248 byte maximum) may be placed in report members. The disk spooling data sets, each containing up to 20 report members, are defined by user supplied and connected IOCBs.

Disk spooling can be interrupted in an orderly manner by using the *stop* function. Spooling can subsequently be continued with the *restart* function. Should an unintentional interruption occur, the disk spooling data set is reinitialized. In this case, the last record residing on the disk can be read to verify data integrity.

Report members may be retrieved more than one time. A specific command is required in order to delete the report members, and a warning occurs if an attempt is made to delete a report member which has not been retrieved in its entirety. As an option, multiple blank characters within text records can be compressed prior to being placed on disk and are expanded to their original form when retrieved. A character other than the EBCDIC blank can be chosen for compression. As an option, the program will assign a sequential record identification number to report member entries.

Format/Print Support

Format/Print Programming RPQ (5799-TBA) provides a method for expanding a variable length record containing EBCDIC character data into a formatted print image.

Format functions are provided which can be used in various combinations to create printer output. The functions allow:

- Move selected fields of variable length data into a user specified storage location
- Merge EBCDIC character data with pre-defined format records to create print images
- Output print images to a printer device

Up to four printers may be connected at one time. When used in conjunction with the Disk Spooling Programming RPQ, support is extended to formatted printing of reports which have been previously stored on disk and retrieved by that program. These two Programming RPQs in combination provide a complete spooling package for the Control Program Support user.

Autocall Support

Autocall Support Programming RPQ (5799-TBC) provides a set of functional modules to be used in conjunction with Control Program Support for the support of the Autocall Originate Hardware. This support consists of a macro routine that the user can code to provide outgoing dialing of a prepared telephone number. The execution code will interface with the Autocall Hardware to dial digits 0–9 and provide for 0-to 15-second pauses between digits. This Programming RPQ will support the two line interfaces on each autocall unit and multiple autocall units per system.

Magnetic Stripe Card Reader

The Magnetic Stripe Card Reader Programming RPQ (5799-TAJ) provides an interface between the application program and the Explosion Proof Card Reader RPQ D02008. The program accepts data from the card readers, translates the data, validates the data and presents it to the application program in a usable form. The card can be read either on insertion or upon withdrawal.

A series of pulses encoded on a magnetic stripe on the card are translated into digits represented in ABA (American Bankers Association) code. Each digit has a unique four bit code. The program support also performs parity checking and LRC (longitudinal redundancy check) verification.

When the end of the string of pulses is sensed, the translated data is made available to the application program. The application program then processes the data associated with a paritcular reader.

Other features include:

- Installation Verfication
- Error Iogging by Pl Attachment card
- Error documentation (SOM, EOM, LRC Counts)
- Single interrupt recognition program
- Online reader range checking (partial PI card)
- Interrupt on positive transition only processing
- 15-Character read capability
- Supports up to 128 readers

Service Support

The Control Program Support extensions that provide service facilities are:

- Operator Station/Debug Support Programming RPQ
- 4978/4979 Display Map Support Programming RPQ

Operator Station/Debug Support

The Operator Station/Debug Programming RPQ (5799-TBB) provides a set of functional modules to be used in conjunction with Control Program Support for providing an operator interface to the application program. The modules allow activation of tasks from either an 4979 Display Station or other operator station. Integrated into this support are debug features (Dump, Patch, Snap, Breakpoint, Trap, Calc, and Mark) and a message transfer function.

Operator Interface Support. This module provides a software interface to execute a preassigned task via a console interrupt and input command from either an operator or display station. The tasks to be executed must be defined by specification macros inserted in the program.

4978/4979 Display Map Support

4978/4979 Display Map Programming RPQ (5799-TBE) provides a set of functional modules to be used with Control Program Support for the generation of display station formats, known as 'Maps' and the handling of data associated with those Maps.

The Display Map Programming RPQ builds upon the basic 4978/4979 Programming RPQ products for developing applications requiring considerable operator guidance for the data entry and inquiry functions. It is composed of:

- An offline utility function, shipped in loadable form, for the definition and maintenance of screen Maps
- A macro library containing specification and execution macros for use in application programs
- Object modules, also for use in application programs, to support these execution macros

Standalone Utilities

System control programming is provided through a set of stand-alone utilities (5719-SC2) furnished with each Series/1. These utilities enable you to IPL, initialize diskettes and disks, copy, dump, patch, perform automatic system build, and verify the Series/1 system. The utilities include:

- *Diskette IPL Bootstrap.* The Diskette IPL Bootstrap utility loads a program one track in length into the high end of storage.
- *IPL Bootstrap/Loader Disk.* The IPL Bootstrap/Loader Disk utility loads programs from the disk into main storage.
- *Diskette Initialization.* The Diskette Initialization utility initializes the diskette, writing ID records and checking for bad cylinders, and assigning alternate cylinders. The program initializes for Basic Exchange Format and formats sectors to 128 bytes.
- *Disk Initialization.* The Disk Initialization utility initializes the disk, writing sector IDs and checking for bad sectors, and assigns alternate sectors. The program provides for user specified alternate sector assignment.
- *Create Diskette HDR1.* The Create Diskette HDR1 creates an HDR1 record on track 0 for a diskette, using specified information.
- *Delete Diskette HDR1.* The Delete Diskette HDR1 utility deletes the HDR1 record for a specified diskette data file.
- *Diskette to Disk Copy.* The Diskette to Disk Copy utility copies data from a specified diskette file to a specified disk file.
- *Disk to Diskette Copy.* The Disk to Diskette Copy utility copies from a specified disk file to a specified diskette file.
- *Diskette to Printer Dump.* The Diskette to Printer Dump utility dumps the contents of a specified area on the diskette to the printer.
- *Disk to Printer Dump.* The Disk to Printer Dump utility dumps the contents of a specified area on the disk to the printer.
- *Operator Station to Diskette Patch.* The Operator Station to Disk Patch utility applies a patch entered at the operator station to a specified location on the disk.
- *Standalone Storage to Diskette Dump.* Pressing the LOAD key on the console loads this utility into main storage from a dedicated prebuilt diskette. The utility then dumps the contents of storage to that same diskette.

Supported Licensed Programs

Control Program Support support is the basis for the following Series/1 licensed programs:

- Structured Programming Facility
  - MVS/VTAM application program
  - MVS/TCAM application program
- Base Program Preparation Facilities—for additional information about this program, contact your IBM representative. Through conversion aids, the Program Preparation Subsystem can also be used.

Structured Programming Facility

Series/1 Structured Programming Facility (5719-ED1) is a programming editor that resides in a Series/1 and communicates with:

- A System/370 MVS/VTAM system through the MVS/VTAM Application Program (5719-CR1), or
- A System/370 MVS/TCAM system through the MVS/TCAM Application Program (5719-CR2), which services requests from the Series/1 Structured Programming Facility

The connection of the Series/1 to the System/370 consists of BSC lines using IBM 3271 protocol.

The types of requests handled are:

- Retrieve data from the System/370 and send it to the Series/1
- Receive data from the Series/1 and store it on System/370 direct access storage
- Submit background jobs to JES2 from the Series/1 or a data set on the System/370 direct access storage
- Request background job status from MVS/JES2 and send the job status to the Series/1

The Series/1 Structured Programming Facility editor provides most of the editing functions which are provided by System/370 TSO/Structured Programming Facility in a format and mode of operation consistent with System/370 Structured Programming Facility. Editing functions provided by Series/1 Structured Programming Facility include the following:

- Full screen context editing which allows multiple lines to be modified in a single transaction
- Forward, backward, and sideways scrolling of data through use of program function keys
- Inserting and deleting of a line or a block of lines
- Moving or copying of a line or a block of lines
- Shifting data on a line or lines

The use of Series/1 Structured Programming Facility does not require any prerequisite programs. The product is distributed as a completely self-contained load module.

Structured Programming Facility MVS/VTAM Application Program

The Series/1 Structured Programming Facility MVS/VTAM Application Program (5719-CR1) provides the software necessary to service requests from Series/1 Structured Programming Facility (5719-ED1), to read and write messages, and to attach service tasks in the System/370 MVS/VTAM system.

This licensed program consists of three System/370-resident programs: MAIN Program, Serivce Program, and Queue Services Program. The MAIN Program reads and writes messages, and attaches multiple Series/1 Structured Programming Facility service tasks.

The Service Program handles the following requests from the Series/1 Structured Programming Facility program resident in the Series/1:

- *Retrieve Data* from the System/370 direct access storage and sent it to the Series/1. Retrievable information includes sequential, member of a partitioned data set, or directory information of a partitioned data set.
- *Store Data* from the Series/1 to System/370 direct access storage. The data from the Series/1 is sequential, and is stored in a System/370 partitioned data set or a sequential data set.
- *Remote Job Entry* request from the Series/1 will write JCL to JES2 from the Series/1 or from an existing member of a partitioned data set or sequential data set.
- *Retrieve Background Job Status* from JES2 by job name or by user ID. This program can handle up to ten such requests at one time.

The Queue Services Program handles the enqueue and dequeue requests for all System/370 functions.

Structured Programming Facility MVS/TCAM Application Program

The Series/1 Structured Programming Facility MVS/TCAM Application Program (5719-CR2) provides the software necessary to service requests from Series/1 Structured Programming Facility (5719-ED1), to read and write messages, and to attach service tasks in the System/370 MVS/TCAM system.

This licensed program consists of three System/370-resident programs: MAIN Program, Service Program, and Queue Services Program. The MAIN Program reads and writes messages, and attaches multiple Series/1 Structured Programming Facility service tasks.

The Service Program handles the following requests from the Series/1 Structured Programming Facility program resident in the Series/1:

- *Retrieve Data* from the System/370 direct access storage and send it to the Series/1. Retrievable information includes sequential, member of a partitioned data set, or directory information of a partitioned data set.
- *Store Data* from the Series/1 to System/370 direct access storage. The data from the Series/1 is sequential, and is stored in a System/370 partitioned data set or a sequential data set.
- *Remote Job Entry* request from the Series/1 will write JCL to JES2 from the Series/1 or from an existing member of a partitioned data set or sequential data set.
- *Retrieve Background Job Status* from JES2 by job name or by user ID. This program can handle up to ten requests at one time.

The Queue Services Program handles the enqueue and dequeue requests for all System/370 functions.

Conversion Aids

The following Programming RPQs provide program preparation conversion support.

Series/1 Base Program Preparation Facilities to Series/1 Program Preparation Subsystem Object Module Conversion Program Aid Programming RPQ. This Programming RPQ converts Base Program Preparation Facilities user-generated object modules into a format compatible with the Program Preparation System and stores them in a Realtime Programming System partitioned data set. It also supports Control Program Support Programming RPQ object modules. Once the modules are converted, the user can create and build his own programs with the converted modules to create Series/1 loads modules.

Series/1 Realtime Programming System Interactive IPL Loader Programming RPQ. This Programming RPQ allows the user to define the load device and the data set name to be loaded at IPL time. It is required in order to interchange loads with the Realtime Programming System, Control Program Support, and/or Event Driven Executive supervisors. It is also required in order to load supervisors from multiple devices.

System Requirements

The minimum system requirements to support system generation are:

- Processor—IBM 4952, 4953, or 4955 Processor with at least 32K bytes of processor storage
- Operator Station—a Teletype* Model ASR 33/35 or equivalent ASCII device
- Disk/Diskette—one IBM 4962 Disk Storage Unit Model 2 or Model 2F (combination disk/diskette unit) or one IBM 4962 Disk Storage Unit Model 1 or Model 1F *and* one IBM 4964 Diskette Unit
- Printer—one IBM 4974 Printer

In addition to the minimum hardware configuration, multiple diskettes and disks are supported.

Note: If Program development support is required for:

- 4962 Disk Storage Unit, Model 3 or 4, or
- 4963 Disk Storage Unit, or
- 4966 Diskette Magazine Unit, or
- 4978 Display Station, or
- 4979 Display Station

use the Realtime Programming System/Program Preparation Subsystem and the following Programming RPQs:

- Series/1 Base Program Preparation Facilities to Series/1 Program Preparation Subsystem Object Module Conversion Program (5799-TCN)
- Series/1 Realtime Programming System Interactive IPL Loader (5799-TCK)

Registered Trademark of Teletype Corporation

Appendix B. Series/1 Programming Publications Directory

This directory contains the books for all of the programming products discussed in the Series/1 Programming System Summary. In addition, it contains Series/1 Programming RPQ books. Hardware books are not listed.

The directory is arranged alphabetically by product name. All books describing a product are listed alphabetically following the product name. Documents withdrawn and no longer available for order are denoted with a "W". All other documents are available as of September 26, 1980.

Book entries following the product name contain the title, the order number, technical newsletter order numbers, and program numbers. A sample entry follows:

```
Realtime Programming System
 Version 4: . . . . . .Product-name
    Operator Commands and
     Utilities SC34-0208  . . . . . .Book title and order number
        and Newsletter
         SN34-0557  . . . . . .Technical Newsletter order number
        Program Number:
        5719-PC4  . . . . . .Program number
    Supervisor Program Logic
     Manual (W) LY34-0116 . . . . . .Book title and order number of
        Program Number:                    a withdrawn document
        5719-PC4  . . . . . .Program number
```

While the utmost care has been taken in developing this directory, errors may exist. If you find any errors, or have any ideas on how we may improve this directory, please, use the form at the back of this book to let us know.

Authorized Program Analysis Report (APAR):
- User's Guide GC34-0099-5
  - Program Number:
    5719-PC1 5719-PC2
    5719-PC3 5719-PC4
    5719-XS1 5719-XS2

Basic Program Preparation Facility to Program Preparation Subsystem Object Module Conversion Programming RPQ P82538:
- Licensed Program Specifications GC34-1621
  - Program Number:
    5719-TCN
- User's Guide SC34-1619
  - Program Number:
    5719-TCN

Basic Program Preparation Facility:
- Licensed Program Design Objectives (W) GC34-0077
  - Program Number:
    5719-PA1
- Macro Assembler Language Reference Summary SX34-0076
  - Program Number:
    5719-PA1
- Macro Assembler Program Logic Manual LY34-0075
  - Program Number:
    5719-PA1
- Macro Assembler Programmer's Guide (W) SC34-0074
  - Program Number:
    5719-PA1
- Program Logic Manual LY34-0073
  - Program Number:
    5719-PA1
- Program Product Specifications GC34-0091
  - Program Number:
    5719-PA1
- User's Guide (W) SC34-0072 and Newsletter (W) SN34-0403
  - Program Number:
    5719-PA1

Control Program Support Address Relocation Translator Support Programming RPQ P82536:
- Licensed Program Specifications GC34-1601
  - Program Number:
    5799-TBT
- User's Guide SC34-1602
  - Program Number:
    5799-TBT

Control Program Support Autocall Support Programming RPQ

P82533:
Program Logic Manual
LY34-0573
Program Number:
5799-TBC
User's Guide SC34-1565
Program Number:
5799-TBC

Control Program Support Binary Synchronous Communication Programming RPQ P82516:
Program Logic Manual
LY34-0558-1
Program Number:
5799-TAF
User's Guide SC34-1553-2
Program Number:
5799-TAF

Control Program Support Commercial Arithmetic Programming RPQ P82534:
Program Logic Manual
LY34-0574
Program Number:
5799-TDB
User's Guide SC34-1566
Program Number:
5799-TBD

Control Program Support Disk Spooling Programming RPQ P82529:
Licensed Program Specifications
GC34-1580
Program Number:
5799-TAY
Program Logic Manual
LY34-0571
Program Number:
5799-TAY
User's Guide SC34-1563
and Newsletter
SN34-1567
Program Number:
5799-TAY

Control Program Support Disk Table of Contents Programming RPQ P82528:
Program Logic Manual
LY34-0569
Program Number:
5799-TAW
User's Guide SC34-1561
and Newsletter
SN34-1565
Program Number:
5799-TAW

Control Program Support Extended Function Programming RPQ P82535:
Licensed Program Specifications
GC34-1586
Program Number:
5799-TBQ
Program Logic Manual
LY34-0578
Program Number:
5799-TBQ
User's Guide SC34-1570-1
Program Number:
5799-TBQ

Control Program Support Extension I Programming RPQ P82825:
Program Logic Manual
LY34-0562
Program Number:
5799-TAL
User's Guide SC34-1558
Program Number:
5799-TAL

Control Program Support Extension II Programming RPQ P82526:
Program Logic Manual
LY34-0563
Program Number:
5799-TAQ
User's Guide SC34-1559
Program Number:
5799-TAQ

Control Program Support Format Print Programming RPQ P82530:
Program Logic Manual
LY34-0572
Program Number:
5799-TBC
User's Guide SC34-1564
and Newsletter
SN34-1568
Program Number:
5799-TBA

Control Program Support Indexed Access Method Programming RPQ P82519:
Program Logic Manual
LY34-0561
Program Number:
5799-TAH
User's Guide SC34-1556-1
and Newsletter
SN34-1596
Program Number:
5799-TAH

Control Program Support Operator Station/Debug Package Programming RPQ P82532:
Logic Manual LY34-0570
Program Number:
5799-TBB
User's Guide SC34-1562
Program Number:
5799-TBB

Control Program Support Programming RPQ P82508:
Licensed Program Specifications
GC34-1557-1
Program Number:
5799-TAA
Program Logic Manual (W)
LY34-0557
Program Number:
5799-TAA
User's Guide SC34-1552-2
Program Number:
5799-TAA

Control Program Support Sort/Merge Programming RPQ P82527:
Licensed Program Specifications
GC34-1583
Program Number:
5799-TAT
Program Logic Manual
LY34-0568
Program Number:
5799-TAT
User's Guide SC34-1560
Program Number:
5799-TAT

Control Program Support 4963 Disk Subsystem Programming RPQ

P82541:
Licensed Program Specifications
GC34-1638
Program Number:
5799-TCZ
User's Guide SC34-1645
Program Number:
5799-TCZ

Control Program Support 4963-4966 Save/Restore Utility Programming RPQ P82539:
Licensed Program Specifications
GC34-1622
Program Number:
5799-TDK
User's Guide SC34-1626
Program Number:
5799-TDK

Control Program Support 4969 Magnetic Tape Support: Programming RPQ P82621
Licensed Program Specifications
GC34-0347
Program Number:
5799-TDW
User's Guide SC34-0348
Program Number:
5799-TDW

Control Program Support 4978/4979 Display Map Programming RPQ P82531:
Licensed Program Specifications
GC34-1585
Program Number:
5799-TBE
Program Logic Manual
LY34-0575
Program Number:
5799-TBE
User's Guide SC34-1567
Program Number:
5799-TBE

Control Program Support 4978/4979 Display Station Programming RPQ P82520:
Licensed Program Specifications
GC34-1575
Program Number:
5799-TAK
Program Logic Manual
LY34-0576
Program Number:
5799-TAK
User's Guide SC34-1568-1
and Newsletter
SN34-1559
and Newsletter
SN34-1564
Program Number:
5799-TAK

Control Program Support 4979 Display Station Programming RPQ P82515:
Program Logic Manual (W)
LY34-0559
Program Number:
5799-TAE
User's Guide SC34-1554-1
and Newsletter
SN34-1563
Program Number:
5799-TAE

Control Program Support 4991-201 Magnetic Stripe Card Reader Programming RPQ P82504:
Licensed Program Specifications
GC34-1574
Program Number:
5799-TAJ
Program Logic Manual
LY34-0560-1
Program Number:
5799-TAJ
User's Guide SC34-1555-1
Program Number:
5799-TAJ

D

Directory Assistance System Programming RPQ P82524, P82025, P82024, P82523:
Licensed Program Specifications
GC34-1600
Program Number:
5099-NO3
Program Reference and Operation Manual (W)
GC34-1588
Program Number:
5099-NO3

E

Event Driven Executive COBOL Compiler & Resident Library:
Licensed Program Specifications
GL23-0012
Program Number:
5719-CB3

Event Driven Executive COBOL Transient Library:
Licensed Program Specifications
GL23-0013
Program Number:
5719-CB4

Event Driven Executive COBOL:
Language Reference
GC34-0234-1
Program Number:
5719-CB1 5719-CB2
5719-CB3 5719-CB4
Programmer's Guide
SL23-0014-1
Program Number:
5719-CB3 5719-CB4

Event Driven Executive Data Collection Interactive Programming RPQ P82600:
Licensed Program Specifications
GC34-1652-2
Program Number:
5799-TDE
User's Guide SC34-1654-2
Program Number:
5799-TDE

Event Driven Executive FORTRAN IV Compiler & Object Support Library:
Licensed Program Specifications
GC34-0306-1
Program Number:

5719-F02

Event Driven Executive FORTRAN IV:
- Language Reference Card SX34-0135
  - Program Number: 5719-F01 5719-F02 5719-F03 5719-F04
- Language Reference GC34-0133-1
  - and Newsletter GN34-0601
  - and Newsletter GN34-0649
  - Program Number: 5719-F01 5719-F02 5719-F03 5719-F04
- User's Guide SC34-0315-1
  - and Newsletter SN34-0637
  - and Newsletter SN34-1626
  - Program Number: 5719-F02

Event Driven Executive Indexed Access Method:
- Internal Design LY34-0189-1
  - Program Number: 5719-AM3
- Licensed Program Specifications GC34-0345
  - Program Number: 5719-AM3

Event Driven Executive Macro Assembler:
- Licensed Program Design Objectives (W) GC34-0321
  - Program Number: 5719-ASA
- Licensed Program Specifications GC34-0308
  - Program Number: 5719-ASA
- Reference GC34-0317
  - Program Number: 5719-ASA

Event Driven Executive Mathematical and Functional Subroutine Library:
- Licensed Program Specifications GC34-0307
  - Program Number: 5719-LM3
- User's Guide SC34-0139-2
  - Program Number: 5719-LM1 5719-LM2 5719-LM3

Event Driven Executive Multiple Terminal Manager Version 1:
- Internal Design LY34-0190
  - and Newsletter LN34-1633
  - Program Number: 5719-MS1
- Licensed Program Specifications GC34-0346
  - Program Number: 5719-MS1

Event Driven Executive Multiple Terminal Manager Version 2:
- Internal Design LY34-0213
  - Program Number: 5719-MS2
- Licensed Program Specifications GC34-1700
  - Program Number: 5719-MS2

Event Driven Executive PL/I Compiler and Resident Library:
- Licensed Program Design Objectives (W) GC34-0322
  - Program Number: 5719-PL5 5719-PL6
- Licensed Program Specifications GC34-0145
  - Program Number: 5719-PL5

Event Driven Executive PL/I Transient Library:
- Licensed Program Design Objectives (W) GC34-0322
  - Program Number: 5719-PL5 5719-PL6
- Licensed Program Specifications GC34-0146
  - Program Number: 5719-PL6

Event Driven Executive PL/I:
- Execution Logic Manual SY34-0149
  - Program Number: 5719-PL5 5719-PL6
- Language Reference GC34-0147
  - and Newsletter GN34-0638
  - Program Number: 5719-PL5 5719-PL6
- Messages SC34-0156
  - and Newsletter SN34-0640
  - Program Number: 5719-PL6
- User's Guide SC34-0148
  - and Newsletter SN34-0639
  - Program Number: 5719-PL5 5719-PL6

Event Driven Executive Sort/Merge:
- Licensed Program Specifications GL23-0015
  - Program Number: 5719-SM2
- Programmer's Guide SL23-0016
  - Program Number: 5719-SM2

Event Driven Executive System/370 Channel Attach:
- Internal Design LY34-0215
  - Program Number: 5719-CX1
- Licensed Program Specifications GC34-1701
  - Program Number: 5719-CX1

Event Driven Executive Version 1:
- Basic Supervisor and Emulator Licensed Program Specifications GC34-0301-1
  - Program Number: 5719-XS1
- Communications and Terminal Application Guide SC34-0316-2
  - Program Number: 5719-LM5 5719-UT3 5719-LM6 5719-MS1 5719-XS2 5719-XX2 5719-XX3 5740-LM2

5740-LM3 5719-UT4
Internal Design LY34-0168-2
Program Number:
5719-ASA 5719-XX3
5719-XS1 5719-XX2
5719-XS2
Language Reference
SC34-0314-2
Program Number:
5719-LM3 5719-AM3
5719-LM5 5719-XX2
5719-LM6 5719-XX3
5719-MS1 5740-LM2
Library Summary GC34-0344
Licensed Program Design Objectives GC34-0320
Program Number:
5719-AM3 5719-SM2
5719-CB3 5719-CB4
5719-MS1
Macro Library/Host Licensed Program Specifications
GC34-0305-2
Program Number:
5719-LM2
Macro Library Licensed Program Specifications GC34-0304-2
Program Number:
5719-LM5
Planning Guide Version 1.1 (W)
GC34-0328
Program Number:
5719-LM2 5719-MS1
5719-XS1 5719-UT3
5719-XX2 5719-LM5
Program Preparation Facility Licensed Program Specifications
GC34-0303-1
Program Number:
5719-XX2
Reference Summary
SX34-0101-2
Program Number:
5719-MS1 5719-AM3
5719-UT3 5719-LM5
5719-UT4 5719-LM6
5719-XS1 5719-XX2
5719-XS2 5719-XX3
5740-LM2 5740-LM3
System Guide SC34-0312-2
Program Number:
5719-LM2 5740-LM3
5719-MS1 5719-AM3
5719-UT3 5719-LM5
5719-UT4 5719-LM6
5719-XS1 5719-XX2
5719-XS2 5719-XX3
Tabs SX34-0030
Program Number:
Utilities, Operator Commands, Program Preparation, Messages and Codes SC34-0313-2
Program Number:
5719-AM3 5740-LM2
5719-LM5 5719-UT3
5719-LM6 5719-UT4
5719-XS1 5719-XX2
5719-XS2 5719-XX3
5740-LM3
Utilities Licensed Program Specifications GC34-0302-1
Program Number:
5719-UT3
Event Driven Executive Version 2:
Basic Supervisor and Emulator Licensed Program Specifications
GC34-0353
Program Number:
5719-XS2
Communications and Terminal Application Guide SC34-0316-2
Program Number:
5719-LM5 5719-UT3
5719-LM6 5719-MS1
5719-XS2 5719-XX2
5719-XX3 5740-LM2
5740-LM3 5719-UT4
Internal Design LY34-0168-2
Program Number:
5719-ASA
5719-XS1 5719-XX2
5719-XS2 5719-XX3
Language Reference
SC34-0314-2
Program Number:
5719-LM5 5719-XX2
5719-LM6 5719-XX3
5719-MS1 5740-LM2
5740-LM3 5719-AM3
Macro Library/Host Licensed Program Specifications
GC34-0357
Program Number:
5740-LM3
Macro Library Licensed Program Specifications GC34-0356
Program Number:
5719-LM6
Program Preparation Facility Licensed Program Specifications
GC34-0355
Program Number:
5719-XX3
Reference Summary
SX34-0101-2
Program Number:
5719-MS1 5719-AM3
5719-UT3 5719-LM5
5719-UT4 5719-LM6
5719-XS1 5719-XX2
5719-XS2 5719-XX3
5740-LM2 5740-LM3
System Guide SC34-0312-2
Program Number:
5719-MS1 5719-AM3
5719-UT3 5719-LM5
5719-UT4 5719-LM6
5719-XS1 5719-XX2
5719-XS2 5719-XX3
5740-LM2 5740-LM3
Tabs SX34-0030
Utilities, Operator Commands, Program Preparation, Messages and Codes SC34-0313-2
Program Number:
5719-AM3 5740-LM2
5719-LM5 5719-UT3
5719-LM6 5719-UT4
5719-XS1 5719-XX2
5719-XS2 5719-XX3
5740-LM3
Utilities Licensed Program Specifications GC34-0354

P

Program Number:
5719-AS1

Program Preparation Subsystem Version 2:

Application Builder Program Logic Manual LY34-0128
Program Number:
5719-AS2

Application Builder User's Guide SC34-0153
and Newsletter SN34-0456
and Newsletter SN34-0485
and Newsletter SN34-0577
Program Number:
5719-AS2

Batch User's Guide SC34-0151-1
and Newsletter SN34-0589
Program Number:
5719-AS2

Introduction GC34-0127
and Newsletter GN34-0441
Program Number:
5719-AS2

Job Stream Processor Program Logic Manual LY34-0126
Program Number:
5719-AS2

Licensed Program Design Objectives GC34-0129-1
Program Number:
5719-AS2

Licensed Program Specifications GC34-0155
Program Number:
5719-AS2

Macro Assembler Program Logic Manual LY34-0124
and Newsletter LN34-0438
Program Number:
5719-AS1

Macro Assembler Reference Summary SX34-0127
Program Number:
5719-AS1

Macro Assembler User's Guide SC34-0124-1
Program Number:
5719-AS1 5719-AS2
5719-AS3

Messages SC34-0154
and Newsletter SN34-0503
and Newsletter SN34-0581
Program Number:
5719-AS2

Text Editor Program Logic Manual LY34-0127
Program Number:
5719-AS2

Text Editor User's Guide SC34-0152-1
and Newsletter SN34-0579
Program Number:
5719-AS2

Program Preparation Subsystem Version 3:

Application Builder Program Logic Manual LY34-0132
Program Number:
5719-AS3

Application Builder User's Guide SC34-0224
and Newsletter SN34-0493
and Newsletter SN34-0576
Program Number:
5719-AS3

Batch User's Guide SC34-0222
and Newsletter SN34-0590
Program Number:
5719-AS3

Introduction GC34-0221
Program Number:
5719-AS3

Job Stream Processor and Generation Program Logic Manual LY34-0130
and Newsletter LN34-0591
Program Number:
5719-AS3

Licensed Program Design Objectives GC34-0220-1
Program Number:
5719-AS3

Licensed Program Specifications GC34-0227
Program Number:
5719-AS3

Macro Assembler Program Logic Manual LY34-0133
Program Number:
5719-AS3

Macro Assembler Reference Summary SX34-0130
Program Number:
5719-AS3

Macro Assembler User's Guide SC34-0124-1
Program Number:
5719-AS1 5719-AS2
5719-AS3

Messages SC34-0225
and Newsletter SN34-0583
Program Number:
5719-AS3

Text Editor Program Logic Manual LY34-0131
Program Number:
5719-AS3

Text Editor User's Guide SC34-0223
and Newsletter SN34-0582
Program Number:
5719-AS3

Program Preparation Subsystem Version 4:

Application Builder Program Logic Manual LY34-0161
Program Number:
5719-AS4

Application Builder User's Guide SC34-0275

and Newsletter
SN34-0596
Program Number:
5719-AS4

Batch User's Guide
SC34-0272-1
Program Number:
5719-AS4

Job Stream Processor and
Generation Program Logic
Manual LY34-0158
Program Number:
5719-AS4

Licensed Program Design
Objectives GC34-0269
Program Number:
5719-AS4

Licensed Program Specifications
GC34-0270-1
Program Number:
5719-AS4

Macro Assembler and Macro
Preprocessor Program Logic
Manual LY34-0160
and Newsletter
LN34-0598
Program Number:
5719-AS4

Macro Assembler Reference
Summary SX34-0128
Program Number:
5719-AS4 5719-ASA

Macro Assembler User's Guide
SC34-0274
and Newsletter
SN34-0595
Program Number:
5719-AS4

Messages SC34-0279
and Newsletter
SN34-0597
Program Number:
5719-AS4

Text Editor Program Logic
Manual LY34-0159
Program Number:
5719-AS4

Text Editor User's Guide
SC34-0273
and Newsletter
SN34-0594
Program Number:
5719-AS4

R

Realtime Programming System
Address Translator Transient
Support Version 1
Programming RPQ P82582:
Licensed Program Specifications
GC34-1604
Program Number:
5799-TBW
User's Guide SC34-1603-2
Program Number:
5799-TBW 5799-TBX
5799-TBY

Realtime Programming System
Address Translator Transient
Support Version 2
Programming RPQ P82584:
Licensed Program Specifications
GC34-1606
Program Number:
5799-TBX
User's Guide SC34-1603-2
Program Number:
5799-TBW 5799-TBX
5799-TBY

Realtime Programming System
Address Translator Transient
Support Version 3
Programming RPQ P82582:
Licensed Program Specifications
GC34-1630
Program Number:
5799-TBY
User's Guide SC34-1603-2
Program Number:
5799-TBW 5799-TBX
5799-TBY

Realtime Programming System Basic
Sort Programming RPQ
P82573:
Licensed Program Specifications
SC34-1597
Program Number:
5799-TBP
Program Logic Manual
LY34-0585
Program Number:
5799-TBP
User's Guide SC34-1596
and Newsletter
SN23-0023
and Newsletter
SN23-0042
Program Number:
5799-TBP

Realtime Programming System COBOL
Compiler & Resident Library
Version 1:
Licensed Program Specifications
GC34-0236
Program Number:
5719-CB1

Realtime Programming System COBOL
Transient Library
Version 1:
Licensed Program Specifications
GC34-0237
Program Number:
5719-CB2

Realtime Programming System COBOL
Version 1:
Introduction GC34-0233-1
Program Number:
5719-CB1 5719-CB2
Programmer's Guide
SC34-0235-1
and Newsletter
SN23-0045
Program Number:
5719-CB1 5719-CB2

Realtime Programming System
COBOL:
Language Reference
GC34-0234-1
and Newsletter
GN23-0044
and Newsletter
GN23-0047
Program Number:
5719-CB1 5719-CB2

5719-CB3 5719-CB4
Realtime Programming System Disk Spooling Version 1 Programming RPQ P82574:
Licensed Program Specifications GC34-1595
Program Number: 5799-TBL
Program Logic Manual LY34-0584
Program Number: 5799-TBL
User's Guide SC34-1594-3 and Newsletter SN34-1620
Program Number: 5719-TCG 5799-TBL 5799-TCH
Realtime Programming System Disk Spooling Version 2 Programming RPQ P82574:
Licensed Program Specifications GC34-1615
Program Number: 5799-TCG
Program Logic Manual LY34-0597
Program Number: 5799-TCG
User's Guide SC34-1594-3 and Newsletter SN34-1620
Program Number: 5799-TBL 5799-TCH 5799-TCG
Realtime Programming System Disk Spooling Version 3 Programming RPQ P82574:
Licensed Program Specifications GC34-1616
Program Number: 5799-TCH
Program Logic Manual LY34-0598
Program Number: 5799-TCH
User's Guide SC34-1594-3 and Newsletter SN34-1620
Program Number: 5799-TBL 5799-TCG 5799-TCH
Realtime Programming System FORTRAN IV Compiler & Object Support Library:
Licensed Program Design Objectives GC34-0130
Program Number: 5719-F01 5719-F03
Licensed Program Specifications GC34-0131
Program Number: 5719-F01
Program Logic Manual LY34-0134 and Newsletter LN34-0429
Program Number: 5719-F01
Realtime Programming System FORTRAN IV Realtime Subroutine Library Version 1:
Licensed Program Specifications GC34-0140
Program Number: 5719-F03
Program Logic Manual LY34-0135
Program Number: 5719-F03
Realtime Programming System FORTRAN IV Realtime Subroutine Library Version 2:
Licensed Program Design Objectives GC34-0143
Program Number: 5719-F04
Licensed Program Specifications GC34-0144
Program Number: 5719-F04
Program Logic Manual LY34-0136
Program Number: 5719-F04
Realtime Programming System FORTRAN IV Version 1:
Introduction GC34-0132-2 and Newsletter GN34-0492
Program Number: 5719-F01 5719-F03 5719-F04
Language Reference Card SX34-0135
Program Number: 5719-F01 5719-F02 5719-F03 5719-F04
Language Reference GC34-0133-1 and Newsletter GN34-0601 and Newsletter GN34-0649
Program Number: 5719-F01 5719-F02 5719-F03 5719-F04
User's Guide SC34-0134-3
Program Number: 5719-F01 5719-F02 5719-F03 5719-F04
Realtime Programming System Indexed Access Method Version 1 Programming RPQ P82570:
Licensed Program Specifications GC34-1591
Program Number: 5799-TBN
Program Logic Manual LY34-0582
Program Number: 5799-TBN
User's Guide SC34-1590-2 and Newsletter SN34-1588
Program Number: 5799-TBN 5799-TCA 5799-TCB
Realtime Programming System Indexed Access Method Version 2 Programming RPQ P82570:
Licensed Program Specifications GC34-1609
Program Number: 5799-TCA
Program Logic Manual (W) LY34-0591

Program Number:
5799-TCA
User's Guide SC34-1590-2
and Newsletter
SN34-1588
Program Number:
5719-TCB
5799-TCA 5799-TBN

Realtime Programming System
Indexed Access Method Version 3
Programming RPQ P82570:
Licensed Program Specifications
GC34-1610
Program Number:
5799-TCB
Program Logic Manual
LY34-0592
Program Number:
5799-TCB
User's Guide SC34-1590-2
and Newsletter
SN34-1588
Program Number:
5799-TCA
5799-TCB 5799-TBN

Realtime Programming System
Indexed Access Method:
Licensed Program Design Objectives GC34-0291
Program Number:
5719-AM1
Licensed Program Specifications
GC34-0292
Program Number:
5719-AM1
Program Logic Manual
LY34-0163
and Newsletter
LN34-1597
Program Number:
5719-AM1
User's Guide SC34-0293
and Newsletter
SN34-1598
Program Number:
5719-AM1

Realtime Programming System
Interactive IPL Loader
Programming RPQ P82537:
Licensed Program Specifications
GC34-1620
Program Number:
5799-TCK
User's Guide SC34-1618
and Newsletter
SN34-1593
Program Number:
5799-TCK

Realtime Programming System
Mathematical and Functional
Subroutine Library
Version 1:
Introduction GC34-0138-2
Program Number:
5719-LM1 5719-LM2
Licensed Program Design
Objectives GC34-0136
Program Number:
5719-LM1
Licensed Program Specifications
GC34-0137
Program Number:
5719-LM1
Program Logic Manual
LY34-0139
Program Number:
5719-LM1
User's Guide SC34-0139-2
Program Number:
5719-LM1 5719-LM2
5719-LM3

Realtime Programming System
Mathematical and Functional
Subroutine Library
Version 2:
Introduction GC34-0138-2
Program Number:
5719-LM1 5719-LM2
Licensed Program Design
Objectives GC34-0141
Program Number:
5719-LM2
Licensed Program Specifications
GC34-0142
Program Number:
5719-LM2
Program Logic Manual
LY34-0155
Program Number:
5719-LM2
User's Guide SC34-0139-2
Program Number:
5719-LM1 5719-LM2
5719-LM3

Realtime Programming System
Multiple Terminal Manager
Programming RPQ P82596:
Licensed Program Specifications
GC34-1657-1
Program Number:
5799-TCY
User's Guide SC34-1658-1
Program Number:
5799-TCY

Realtime Programming System PL/I
Compiler & Resident
Library Version 1:
Licensed Program Specifications
GC34-0090
Program Number:
5719-PL1

Realtime Programming System PL/I
Compiler & Resident
Library Version 2:
Licensed Program Design Objectives (W) GC34-0280
Program Number:
5719-PL2 5719-PL4
Licensed Program Specifications
GC34-0281
Program Number:
5719-PL2

Realtime Programming System PL/I
Transient Library Version 1:
Licensed Program Specifications
GC34-0092
Program Number:
5719-PL3

Realtime Programming System PL/I
Transient Library Version 2:
Licensed Program Design Objectives (W) GC34-0280
Program Number:
5719-PL2 5719-PL4
Licensed Program Specifications
GC34-0282
Program Number:
5719-PL4

Realtime Programming System PL/I
Version 1:
Execution Logic Manual
SY34-0086-2
Program Number:
5719-PL1 5719-PL3
5719-PL2 5719-PL4
Introduction GC34-0084
Program Number:
5719-PL1 5719-PL3
Language Reference
GC34-0085-2
Program Number:
5719-PL1 5719-PL3
5719-PL2 5719-PL4
Licensed Program Design
Objectives (W) GC34-0083
Program Number:
5719-PL1 5719-PL3
Messages SC34-0088-1
and Newsletter
SN34-0632
Program Number:
5719-PL3 5719-PL4
User's Guide SC34-0086-2
and Newsletter
SN34-0623
Program Number:
5719-PL1 5719-PL3
5719-PL2 5719-PL4

Realtime Programming System PL/I
Version 2:
Execution Logic Manual
SY34-0086-2
Program Number:
5719-PL2 5719-PL4
Language Reference
GC34-0085-2
Program Number:
5719-PL1 5719-PL3
5719-PL2 5719-PL4
Messages SC34-0088-1
and Newsletter
SN34-0632
Program Number:
5719-PL3 5719-PL4
User's Guide SC34-0086-2
and Newsletter
SN34-0623
Program Number:
5719-PL1 5719-PL3
5719-PL2 5719-PL4

Realtime Programming System
Presentation Support/Screen
Design Aid Utility:
Licensed Program Design Objectives GC34-0325
Program Number:
5719-SF2 5719-SF1
Licensed Program Specifications
GC34-0326
Program Number:
5719-SF2 5719-SF1
User's Guide SC34-0327
Program Number:
5719-SF1 5719-SF2

Realtime Programming System
Presentation Support:
Program Logic Manual
LY34-0177
Program Number:
5719-SF2

Realtime Programming System Remote
Management Utility
Programming RPQ P82597:
Licensed Program Specifications
GC34-1663
Program Number:
5799-TDH
User's Guide SC34-1664
Program Number:
5799-TDH

Realtime Programming System Screen
Design Aid Utility:
Program Logic Manual
LY34-0176
Program Number:
5719-SF1

Realtime Programming System
Sort/Merge Version 1:
Licensed Program Specifications
GL23-0010
Program Number:
5719-SM1

Realtime Programming System
System/370 Channel Attach
Program:
General Information Manual
GC34-0217
Program Number:
5719-CA1
Licensed Program Design
Objectives GC34-0213
Program Number:
5719-CA1
Licensed Program Specifications
GC34-0214
Program Number:
5719-CA1
Program Logic Manual
LY34-0129
Program Number:
5719-CA1
Reference Manual SC34-0215-1
Program Number:
5719-CA1
Reference Summary SX34-0029
Program Number:
5719-CA1

Realtime Programming System
Transient Activity Tool
Programming RPQ P82606:
Licensed Program Specifications
GC34-0286
Program Number:
5799-TDG
User's Guide SC34-0287
Program Number:
5799-TDG

Realtime Programming System
Version 1:
Communications Program Logic
Manual LY34-0105
Program Number:
5719-PC1
Communications User's Guide
SC34-0105-1
Program Number:
5719-PC1
Data Management Program Logic
Manual LY34-0104
Program Number:
5719-PC1
Data Management User's Guide
SC34-0104-1
Program Number:
5719-PC1
Generation and Installation

Procedures SC34-0110-1
Program Number:
5719-PC1
Introduction GC34-0102-2
Program Number:
5719-PC1
Licensed Program Design
Objectives GC34-0100
Program Number:
5719-PC1
Licensed Program Specifications
GC34-0101
Program Number:
5719-PC1
Macro Reference SC34-0106
Program Number:
5719-PC1
Messages and Codes SC34-0109
Program Number:
5719-PC1
Operation and Support Features
Program Logic Manual
LY34-0107
Program Number:
5719-PC1
Operator Commands and Utilities
SC34-0107
Program Number:
5719-PC1
Problem Determination and
Control Blocks SC34-0111
Program Number:
5719-PC1
Service Release Guide Program
Logic Manual LY34-0120
Program Number:
5719-PC1
Service Release Guide
SC34-0212
and Newsletter
SN34-0487
Program Number:
5719-PC1
Supervisor Program Logic
Manual LY34-0103
Program Number:
5719-PC1
Supervisor User's Guide
SC34-0103-1
Program Number:
5719-PC1

Realtime Programming System
Version 2:
Communications Program Logic
Manual (W) LY34-0110
and Newsletter (W)
LN34-0473
Program Number:
5719-PC2
Communications User's Guide
SC34-0165
and Newsletter
SN34-0471
Program Number:
5719-PC2
Data Management Program Logic
Manual (W) LY34-0109
and Newsletter (W)
LN34-0466
Program Number:
5719-PC2
Data Management User's Guide
SC34-0164-1
Program Number:
5719-PC2
Functional Flow LY34-0152
Program Number:
5719-PC2
Generation and Installation
Procedures SC34-0162-1
Program Number:
5719-PC2
Introduction and Planning Guide
GC34-0102-1 Supplement
GC34-0160
Program Number:
5719-PC1 5719-PC2
Introduction GC34-0114-1
Program Number:
5719-PC2
Licensed Program Design
Objectives GC34-0112-1
Program Number:
5719-PC2
Licensed Program Specifications
GC34-0168
Program Number:
5719-PC2
Macro Reference SC34-0169
and Newsletter
SN34-0463
Program Number:
5719-PC2
Messages and Codes SC34-0167
and Newsletter
SN34-0462
Program Number:
5719-PC2
Operation and Support Features
Program Logic Manual (W)
LY34-0111
and Newsletter (W)
LN34-0472
Program Number:
5719-PC2
Operator Commands and Utilities
SC34-0166
and Newsletter
SN34-0461
Program Number:
5719-PC2
Problem Determination and
Control Blocks SC34-0170
and Newsletter
SN34-0464
Program Number:
5719-PC2
Supervisor Program Logic
Manual (W) LY34-0108
and Newsletter (W)
LN34-0465
Program Number:
5719-PC2
Supervisor User's Guide
SC34-0163
and Newsletter
SN34-0453
Program Number:
5719-PC2
Utilities Program Logic Manual
(W) LY34-0121
and Newsletter (W)
LN34-0467
Program Number:
5719-PC2

Realtime Programming System
Version 3:
Communications Macro Program-

mer's Guide SC34-0197
and Newsletter
SN34-0494
and Newsletter
SN34-0528
and Newsletter
SN34-1612
Program Number:
5719-PC3

Communications Program Logic Manual (W) LY34-0114
Program Number:
5719-PC3

Control Blocks SC34-0218
and Newsletter
SN34-0531
and Newsletter
SN34-1617
Program Number:
5719-PC3

Data Management Macro Programmer's Guide SC34-0196
and Newsletter
SN34-0495
and Newsletter
SN34-0526
and Newsletter
SN34-1611
Program Number:
5719-PC3

Data Management Program Logic Manual (W) LY34-0113
Program Number:
5719-PC3

Design Guide SC34-0191
and Newsletter
SN34-1608
Program Number:
5719-PC3

Functional Flow LY34-0153
and Newsletter
LN34-1618
Program Number:
5719-PC3

Generation and Installation Procedures SC34-0194
and Newsletter
SN34-0502
and Newsletter
SN34-0529
and Newsletter
SN34-1609
Program Number:
5719-PC3

Introduction GC34-0193
Program Number:
5719-PC3

Licensed Program Design Objectives GC34-0115-1
Program Number:
5719-PC3

Licensed Program Specifications GC34-0192
Program Number:
5719-PC3

Macro Reference SC34-0201
and Newsletter
SN34-0497
and Newsletter
SN34-0523
and Newsletter
SN34-1616
Program Number:
5719-PC3

Messages and Codes SC34-0199
and Newsletter
SN34-0499
and Newsletter
SN34-0527
and Newsletter
SN34-1614
Program Number:
5719-PC3

Operation and Support Features Program Logic Manual (W) LY34-0115
Program Number:
5719-PC3

Operator Commands and Utilities SC34-0198
and Newsletter
SN34-0498
and Newsletter
SN34-0520
and Newsletter
SN34-0525
and Newsletter
SN34-1613
Program Number:
5719-PC3

Problem Determination SC34-0200
and Newsletter
SN34-0500
and Newsletter
SN34-0530
and Newsletter
SN34-1615
Program Number:
5719-PC3

Supervisor Macro Programmer's Guide SC34-0195
and Newsletter
SN34-0524
and Newsletter
SN34-1610
Program Number:
5719-PC3

Supervisor Program Logic Manual (W) LY34-0112
Program Number:
5719-PC3

Utilities Program Logic Manual (W) LY34-0138
Program Number:
5719-PC3

Realtime Programming System Version 4:

Binary Synchronous and Start-Stop Communications Macro Programmers Guide SC34-0207-1
Program Number:
5719-PC4

Binary Synchronous/Start-Stop Communications Program Logic Manual (W) LY34-0118
Program Number:
5719-PC4

Command Language Facility Commands Reference Summary SX34-0026-1
Program Number:
5719-PC4

Command Language Facility Language Reference Summary SX34-0027-1

Program Number:
5719-PC4
Command Language Facility
Program Logic Manual (W)
LY34-0165
and Newsletter (W)
LN34-0586
Program Number:
5719-PC4
Command Language Facility
User's Guide SC34-0299-1
Program Number:
5719-PC4
Control Blocks SC34-0210-1
and Newsletter
SN34-1625
Program Number:
5719-PC4
Data Management Macro
Programmer's Guide SC34-0206
and Newsletter
SN34-0553
Program Number:
5719-PC4
Data Management Program Logic
Manual (W) LY34-0117
Program Number:
5719-PC4
Design Guide SC34-0242
Program Number:
5719-PC4
Functional Flow LY34-0154-1
Program Number:
5719-PC4
Generation and Installation
Procedures SC34-0204-2
Program Number:
5719-PC4
Introduction GC34-0114
Supplement GC34-0161-1
Program Number:
5719-PC4
Licensed Program Design
Objectives GC34-0113-3
Program Number:
5719-PC4
Licensed Program Specifications
GC34-0202-1
Program Number:
5719-PC4
Macro Reference SC34-0211-1
and Newsletter
SN34-0555
and Newsletter
SN34-1628
Program Number:
5719-PC4
Messages and Codes SC34-0209
and Newsletter
SN34-0556
Program Number:
5719-PC4
Operation and Support Features
Program Logic Manual (W)
LY34-0119
Program Number:
5719-PC4
Operator Commands and Utilities
SC34-0208
and Newsletter
SN34-0557
Program Number:
5719-PC4
Preliminary Programmer's Guide
(W) SC34-0229
Program Number:
5719-PC4
Problem Determination
SC34-0219
and Newsletter
SN34-0560
and Newsletter
SN341634
Program Number:
5719-PC4
Supervisor Macro Programmer's
Guide SC34-0205
and Newsletter
SN34-0552
and Newsletter
SN34-1627
Program Number:
5719-PC4
Supervisor Program Logic
Manual (W) LY34-0116
Program Number:
5719-PC4
Systems Network Architecture
Support Installation Guide
SC34-0342
Program Number:
5719-PC4
Systems Network Architecture
Support Macro Programmer's
Guide SC34-0228
and Newsletter
SN34-0561
Program Number:
5719-PC4
Systems Network Architecture
Support Program Logic
Manual (W)
LY34-0156
Program Number:
5719-PC4
Utilities Program Logic Manual
(W) LY34-0151
Program Number:
5719-PC4
Realtime Programming System Visual
Information Presentation
Support Programming RPQ
P82595:
Licensed Program Specifications
GC34-1647
Program Number:
5799-TCL
User's Guide SC34-1648
Program Number:
5799-TCL
Realtime Programming System 4969
Magnetic Tape Subsystem
Support Program:
Licensed Program Design Objec-
tives GC34-0288-1
Program Number:
5719-TA4
Licensed Program Specifications
GC34-0289
Program Number:
5719-TA4
Program Logic Manual
LY34-0162
Program Number:
5719-TA4
User's Guide SC34-0290
Program Number:
5719-TA4

Realtime Programming System 4978
Display Station Support
Version 1 Programming
RPQ P82572:
Licensed Program Specifications
GC34-1593
Program Number:
5799-TBM
Program Function Key
Assignment Sheet GX34-0032
Program Logic Manual
LY34-0583
Program Number:
5799-TBM
User's Guide SC34-1592-2
and Newsletter
SN34-0507
and Newsletter
SN34-1600
Program Number:
5799-TBM 5799-TCD
5799-TCE

Realtime Programming System 4978
Display Station Support
Version 2 Programming RPQ
P82572:
Licensed Program Specifications
GC34-1612
Program Number:
5799-TCD
Program Function Key
Assignment Sheet GX34-0032
Program Logic Manual
LY34-0594
Program Number:
5799-TCD
User's Guide SC34-1592-2
and Newsletter
SN34-0507
and Newsletter
SN34-1600
Program Number:
5799-TCD 5799-TBM
5799-TCE

Realtime Programming System 4978
Display Station Support
Version 3 Programming RPQ
P82572:
Licensed Program Specifications
GC34-1613
Program Number:
5799-TCE
Program Function Key
Assignment Sheet GX34-0032
Program Logic Manual
LY34-0595
Program Number:
5799-TCE
User's Guide SC34-1592-2
and Newsletter
SN34-0507
and Newsletter
SN34-1600
Program Number:
5799-TCD
5799-TCE 5799-TBM

Realtime Programming System 5250
Information Display System
Attachment Support:
Licensed Program Design Objectives GC34-0243
Program Number:
5719-TA1
Licensed Program Specifications
GC34-0245
Program Number:
5719-TA1
Program Logic Manual
LY34-0157
Program Number:
5719-TA1
Reference Summary GX34-0036
Program Number:
5719-TA1
User's Guide and Reference
Manual SC34-0246
Program Number:
5719-TA1

Remote Job Entry Programming RPQ
P82575:
Licensed Program Specifications
GC34-1599
Program Number:
5799-TBK
Program Logic Manual
LY34-0581
and Newsletter
LN34-0585
Program Number:
5719-TBK
User's Guide SC34-1589
and Newsletter
SN34-1562
and Newsletter
SN34-1595
Program Number:
5799-TBK

S

Series/1:
Programming System Summary
GC34-0285-1
System Summary
GA34-0035-5

Stand Alone Utilities:
Program Logic Manual (W)
GY34-0071
and Newsletter (W)
GN34-0402
Program Number:
5719-SC2
System Control Program
Specifications GC34-0066
Program Number:
5719-SC2
User's Guide (W) GA34-0070
Program Number:
5719-SC2

Structured Programming Facility
MVS/VTAM Application
Program Version 1:
Licensed Program Design Objectives GC34-1632
Program Number:
5719-CR1
Licensed Program Specifications
GC34-1636-2
Program Number:
5719-CR1

Structured Programming Facility
MVS/VTAM Application
Program Version 2:
Licensed Program Design Objectives GC34-1633

Program Number:
5719-CR2
Licensed Program Specifications
GC34-1637-2
Program Number:
5719-CR2
Structured Programming Facility:
Licensed Program Design Objectives GC34-1631
Program Number:
5719-ED1
Licensed Program Specifications
GC34-1623-2
Program Number:
5719-ED1
Reference Summary GX34-0031
Program Number:
5719-CR1 5719-CR2
5719-ED1
User's Guide SC34-1627-1
Program Number:
5719-CR1 5719-CR2
5719-ED1

T

Telephone Applications Driver Programming RPQ P82567 and P82614:
Program Reference and Operation Manual SC34-1587-2
Program Number:
5799-TBG 5799-TDL
Telephone Applications Driver Programming RPQ P82567:
Licensed Program Specifications
GC34-1598-2
Program Number:
5799-TBG
Telephone Applications Driver Programming RPQ P82614:
Licensed Program Specifications
GC34-1665
Program Number:
5799-TDL

Numeric Subjects

4987 Programmable Communication Subsystem Execution Support:
Licensed Program Design Objectives GC34-0181
Program Number:
5719-CS1
Licensed Program Specifications
GC34-0183
Program Number:
5719-CS1
Program Logic Manual
LY34-0180
Program Number:
5719-CS1
Reference SC34-0180-1
Program Number:
5719-CS1
4987 Programmable Communication Subsystem Extended Execution Support:
General Information Manual
GC34-0186
Program Number:
5719-CS2
Licensed Program Design Objectives GC34-0184
Program Number:
5719-CS2
Licensed Program Specifications
GC34-0185
Program Number:
5719-CS2
Program Logic Manual
LY34-0181
Program Number:
5719-CS2
Reference SC34-0187 and Newsletter SN34-1619
Program Number:
5719-CS2
4987 Programmable Communication Subsystem Preparation Facility:
Licensed Program Design Objectives GC34-0118
Program Number:
5719-CS0
Licensed Program Specifications
GC34-0182
Program Number:
5719-CS0
Reference SC34-0119-2
Program Number:
5719-CS0
523X Entry Station Direct Attachment Support Programming $RPQ P82600:
Licensed Program Design Objectives GC34-1653
Program Number:
5799-TDE

Index

READER'S COMMENT FORM

GC34-0285-1

IBM Series/1 Programming System
Summary

Your comments assist us in improving the usefulness of our publications; they are an important part of the input used in preparing updates to the publications. 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.

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

Corrections or clarifications needed:

Page Comment

Cut or Fold Along Line

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

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Reader's Comment Form

Cut Along Line

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Systems Publications, Dept 27T
P.O. Box 1328
Boca Raton, Florida 33432

Fold and tape | Please Do Not Staple | Fold and tape

IBM®

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

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

GC34-0285-1
Printed in U.S.A.

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
(International)

GC34-0285-1
Printed in U.S.A.