GH20-1028-3

Program Product

Customer Information Control System (CICS) General Information Manual

IBM

GH20-1028-3

Program Product

Customer Information Control System (CICS) General Information Manual

Program Nos. DOS-ENTRY 5736-XX6
DOS-STANDARD 5736-XX7
OS-STANDARD V2 5734-XX7

The IBM Customer Information Control System (CICS) is a transaction-oriented, multiapplication data base/data communication interface between a System/360 or System/370 operating system and user-written application programs. Applicable to most online systems, CICS provides many of the facilities necessary for standard terminal applications: message switching, inquiry, data collection, order entry, and conversational data entry.

CICS is available in three systems – two for DOS users and one for OS users. Because the two CICS/DOS systems are compatible with each other and with the CICS/OS system, it is possible to start with a small DOS data base/data communication configuration and move up through DOS into OS.

This manual is designed for persons interested in a general description of CICS and its data base/data communication capabilities.

IBM

PREFACE

This publication is a general description of three IBM program products: CICS/DOS-ENTRY, CICS/DOS-STANDARD, and CICS/OS-STANDARD V2. It provides customer executives, system administrators, system analysts, system programmers, application programmers, and terminal operators with a fundamental understanding of CICS. For a more detailed description of CICS, see the following IBM publications:

Application Programmer's Reference Manual (SH20-1047)
System Programmer's Reference Manual (SH20-1043)
Terminal Operator's Guide (SH20-1044)
Operations Guide (CICS/DOS) (SH20-1034)
Operations Guide (CICS/OS) (SH20-1048)
Logic Manual (CICS/DOS-ENTRY) (LY20-0712)
Logic Manual (CICS/DOS-STANDARD) (LY20-0713)
Logic Manual (CICS/OS-STANDARD V2) (LY20-0714)

The words "transaction" and "task" have the same connotation in CICS and are used interchangeably throughout this publication; the processing of a transaction may involve the execution of one or more "programs".

All references to CICS/OS and CICS/OS-STANDARD in this publication are references to the CICS/OS-STANDARD V2 system.

Note: Information in this publication concerning TCAM and 3735 support is applicable only to the CICS/OS system.

Fourth Edition (December 1972)

This edition is a major revision obsoleting GH20-1028-2.

This edition applies to Version 1, Modification Level 1, of the CICS/DOS-ENTRY (5736-XX6) and CICS/DOS-STANDARD (5736-XX7) program products and to Version 2, Modification Level 3, of the CICS/OS-STANDARD (5734-XX7) program product; it also applies to all subsequent versions and modifications unless otherwise indicated in new editions or Technical Newsletters.

Changes are continually made to the information herein. Therefore, before using this publication, consult the latest System/360 and System/370 SRL Newsletter (GN20-0360) for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments has been provided at the back of this publication. If this form has been removed, address comments to: IBM Corporation, Technical Publications Department, 1133 Westchester Avenue, White Plains, New York 10604. Comments become the property of IBM.

CONTENTS

INTRODUCTION

In the conventional batch processing environment, the application programmer plans a series of runs to edit batches of input transactions, update master files, and write output reports. To optimize total run time and streamline the cycle, he must concentrate on careful manipulation of data. In accomplishing this, the data becomes intricately tied to his program logic and is of little value to other applications.

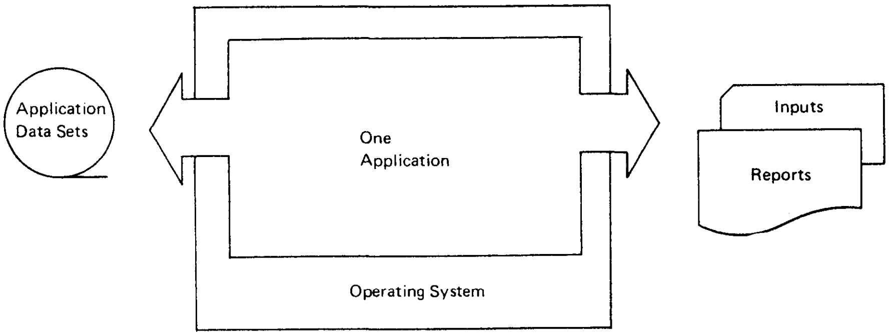


Figure 1. Conventional batch processing

The real-time data base/data communications (DB/DC) environment differs from the conventional batch processing environment primarily in the amount and types of concurrent activities that are likely to occur within the system at a given time. Whereas a batch processing system schedules each application independently and provides data support unique to each application, a DB/DC system controls many transactions arriving on a random nonscheduled basis and provides an integrated data base supporting each application.

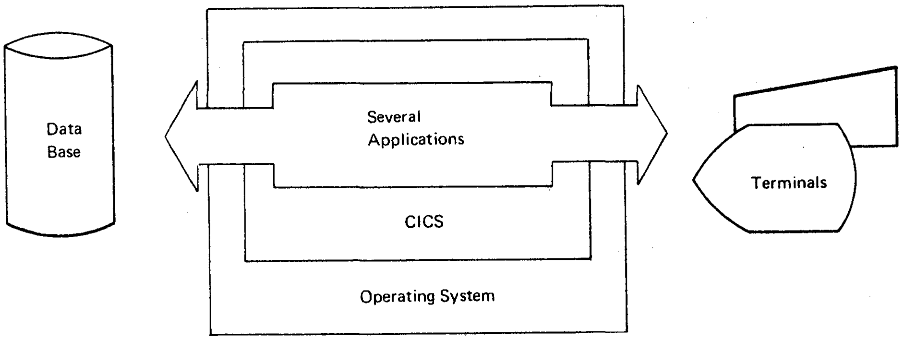


Figure 2. Transaction processing of CICS

In the past, the successful systems have been known as:

- Online information systems
- Real-time informational systems
- Teleprocessing systems
- Data base/data communication systems

These systems required the user to develop a control system that would:

- Host a telecommunication network of mixed devices
- Concurrently manage a wide mixture of transactions being serviced by a variety of programs
- Provide effective controlled access to the data base
- Effectively manage resources, such as main storage, to keep the system in continuous operation
- Prioritize the use of the processing facility
- Provide other real-time facilities necessary for the support of the applications and the environment
- Provide the ancillary system service functions necessary for the successful implementation of data base/data communication systems
- Provide rapid response to the terminals

CICS solves many of these complexities for the application programmer by managing data centrally (in a data base) on behalf of all applications. This shifts the burden of system management considerations from the application programmer to the system programmer and allows the application programmer to concentrate instead on the application.

A key consideration in the selection of a data base/data communication system is that it be appropriate for today's needs and have the growth potential that characterizes the DB/DC environment. CICS is intended to address precisely that consideration; that is, CICS is a family of systems that provides a DB/DC interface to the IBM System/360 and System/370 at most levels of the product line, providing a clearly visible growth or migration path as the user's environment dictates.

Figure 3 shows how the CICS data base supports the information needs of each application, independently and concurrently.

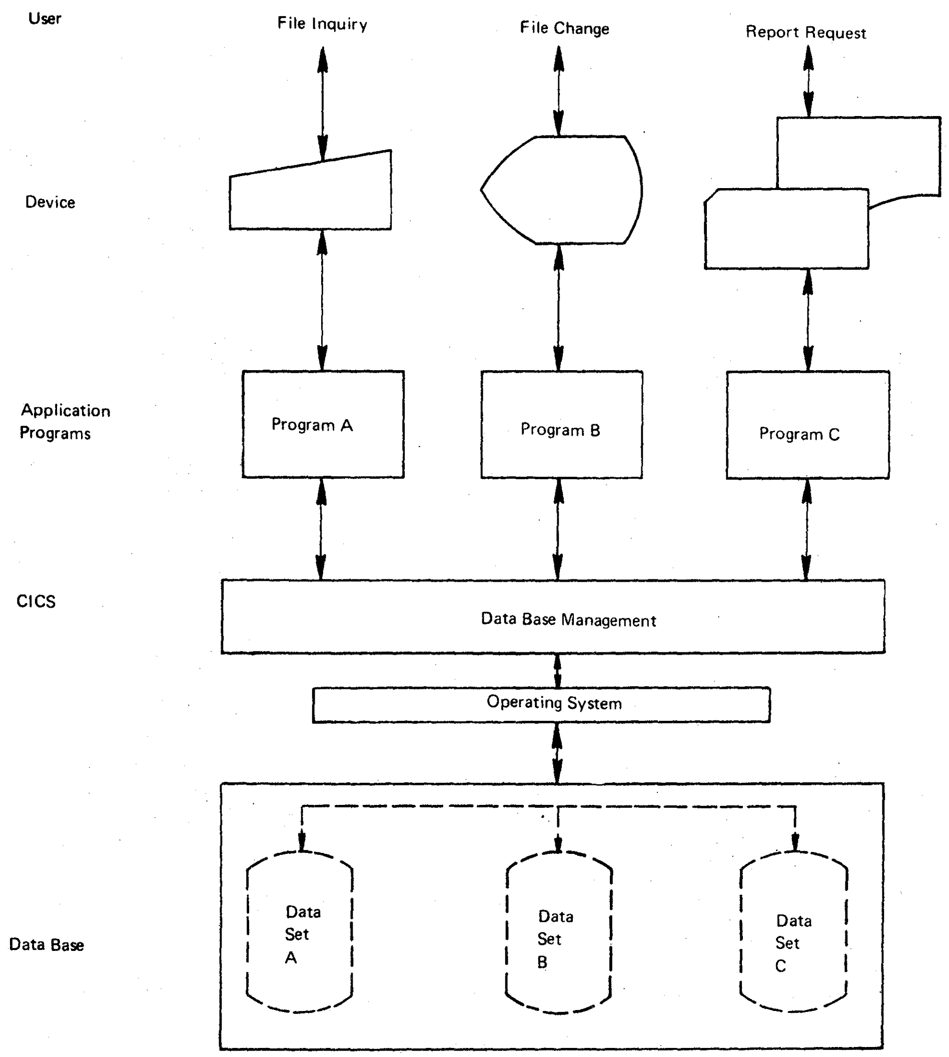


Figure 3. CICS data base concept

GENERAL DESCRIPTION

CICS is a transaction-oriented, multiapplication data base/data communication interface between a System/360 or System/370 operating system and user-written application programs (written in Assembler language, ANS COBOL, or PL/I). Applicable to most online systems, CICS provides many of the facilities necessary for standard terminal applications: message switching, inquiry, data collection, order entry, and conversational and batched data entry.

Functions needed to support a data base/data communication system and standard terminal applications are provided through the following CICS management facilities:

- Task Management
- Storage Management
- Program Management
- Program Interrupt Management
- Time Management
- Dump Management
- Terminal Management
- File Management
- Transient Data Management
- Temporary Storage Management

In addition to these management facilities, CICS provides system service programming to identify terminal operators, to give dynamic control of the entire system to a master terminal, to display real-time system statistics, to intercept abnormal conditions not handled directly by the operating system, and to end operation by gathering summary statistics, closing data sets, and returning control to the operating system.

CICS provides user exits within the CICS management programs that may be selectively generated during the system generation process. For a description of the exits provided, see the section, "Real-Time Management Functions".

MANAGEMENT FACILITIES

TASK MANAGEMENT: Provides the dynamic multitasking facilities necessary for effective, concurrent transaction processing. Functions associated with this facility include priority scheduling, transaction synchronization, and control of serially reusable resources.

STORAGE MANAGEMENT: Controls main storage allocated to CICS. Storage acquisition, disposition, initialization, and request queuing are among the services and functions performed by this component of CICS.

PROGRAM MANAGEMENT: Provides a multiprogramming capability through dynamic program management while offering a real-time program fetch capability. Languages supported are Assembler language, ANS COBOL, and PL/I.

PROGRAM INTERRUPT MANAGEMENT: Provides for the interception of program interrupts by CICS to prevent total system termination. Individual transactions that result in a program check are terminated by CICS with a dump (if Dump Management is used), thus preventing the entire CICS partition/region from terminating. Under CICS/OS, supports the runaway task control function of CICS Time Management.

TIME MANAGEMENT: Provides control of various optional task functions (system stall detection, runaway task control, task synchronization, etc.) based on specified intervals of time or the time of day.

DUMP MANAGEMENT: Provides a facility to assist in analysis of programs and transactions undergoing development or modification. Specified areas of main storage are dumped onto a sequential data set, either tape or disk, for subsequent offline formatting and printing using a CICS utility program.

TERMINAL MANAGEMENT: Provides polling according to user-specified line traffic control as well as user requested reading and writing. This facility supports automatic task initiation to process new transactions. Optionally, the CICS/OS user can request that certain lines be under control of TCAM rather than BTAM. In this case, polling and other network control functions are performed by the user-written TCAM Message Control program (MCP) which resides in a different partition/region. The testing of application programs is accommodated by the simulation of terminals through sequential devices such as card readers, line printers, disk, tape, etc.

FILE MANAGEMENT: Provides a data base facility using direct access and indexed sequential data management. This function supports updates, additions, and random retrieval, and selective retrieval (browsing) of logical data on the data base.

TRANSIENT DATA MANAGEMENT: Provides the optional queuing facility for the management of data in transit to and from user defined destinations. This function has been included to facilitate message switching, data collection, and logging.

TEMPORARY STORAGE MANAGEMENT: Provides the optional general purpose "scratch pad" facility. This facility is intended for video display paging, broadcasting, data collection suspension, conservation of main storage, retention of control information, etc.

SYSTEM CONTROL TABLES

CICS provides the means by which the user describes his data base/data communication environment and the treatment he wishes given to the elements of that environment. The system control tables, which are individually prepared by the user through symbolic description statements, include the following.

PROGRAM CONTROL TABLE: Identifies all valid transactions to be invoked within the system and describes additional transaction attributes.

PROCESSING PROGRAM TABLE: Identifies all valid programs to be executed within the system and describes additional program attributes.

TERMINAL CONTROL TABLE: Describes the terminal environment and the treatment that elements of that environment are to receive.

FILE CONTROL TABLE: Describes the data set (file) environment of the data base and identifies other supplementary service information.

DESTINATION CONTROL TABLE: Describes the transient data environment and treatment to be given to that environment.

SYSTEM SERVICE PROGRAMS

In addition to the management functions described, CICS provides the following system service programs.

SIGN ON/SIGN OFF: Provides terminal operator identification (security).

MASTER TERMINAL FUNCTION: Provides dynamic user control of the system. The master terminal operator can change the status and values of parameters used by CICS and thereby alter the operation of the system.

SUPERVISORY TERMINAL FUNCTION: Performs a terminal-oriented subset of the services available to the master terminal. They are limited to the terminals under a given supervisor's jurisdiction.

SYSTEM STATISTICS: Provides the capability to dynamically log system statistics.

ABNORMAL CONDITION: Intercepts abnormal conditions not handled directly by the operating system.

SYSTEM TERMINATION: Allows the user to terminate operation of CICS by gathering summary statistics, closing data sets, and returning control to the operating system.

ASYNCHRONOUS TRANSACTION PROCESSING: Provides the capability of entering large amounts of data from high-speed terminals, having that data processed asynchronously with other terminal activity, and then having the generated output transmitted back to the same terminal or another terminal.

TRACE: Provides a program debugging facility that reflects the execution of CICS macro instructions by CICS management programs and user-written application programs.

DYNAMIC OPEN/CLOSE: Allows the user to dynamically open/close his data sets during the real-time execution of CICS.

TIME OF DAY CONTROL: Provides the capability for CICS to operate on a round-the-clock basis. CICS adjusts the expiration times it maintains in response to changes in the time of day maintained by the operating system, and then resets its date and time of day to the date and time of day maintained by the operating system.

FUNCTIONAL APPLICATIONS

Data base/data communication systems are applicable to a wide range of uses. Some of the functional applications that are most frequently encountered include the following.

INQUIRY: Selective prestructured retrieval of information from a data base (master file), and is characterized by a small amount of keyed input and a small to moderate output returned to the requesting terminal.

BROWSING: Special case of inquiry where the terminal operator does not have a complete or explicit key to the desired data, and is characterized by selectively displaying a generic portion of the data base until the desired information is found.

ORDER ENTRY: Name frequently associated with the class of applications in which data base oriented information is interactively or conversationally entered from terminals, and is characterized by extensive immediate validation (editing).

DATA ENTRY: Essentially the replacement of keypunch operations by direct entry of keyed data from terminals into the computing system, thereby reducing the exposure to transcription errors. It is characterized by high volume and limited editing.

DATA COLLECTION: Entry of data usually from terminals not key-driven that may or may not interact immediately with the data base. Data collection is characterized by moderate editing and batch control.

MESSAGE SWITCHING/BROADCASTING: The routing of information entered from any valid terminal to one or more destinations (terminals without data base interaction). Message switching/broadcasting is characterized by, but not limited to, administrative (text type) data.

ORDER DISTRIBUTION: The name frequently associated with the application where data base oriented information entered at one terminal on the system triggers the distribution of (potentially) different information to one or more hard-copy terminals. Order distribution is characterized by the application logic (program) which, for example, routes a shipping order to the most appropriate location where the inventory is available. This application is almost invariably an extension of Order Entry.

DATA PROCESSING APPLICATIONS

Data processing applications that employ all or some of the functional applications include:

- Inventory Management
- Production Control
- Maintenance Control
- Purchase Orders
- Service Orders
- Trouble Reporting
- Customer Service
- Engineering Data
- Student Records and Accounting
- Warehouse Management
- Shipping Orders
- Accounts Receivable
- Sales Analysis
- Patient Accounting
- Customer Accounting
- Policy Accounting
- Subscriber Accounting
- Payroll Accounting
- General Accounting
- Stockholder Accounting
- Tax Accounting
- Plant Accounting
- Stock Movement
- Credit Inquiry
- Library Records
- Welfare Accounting
- Police Information
- Waybill Accounting
- Car Accounting
- Reservations Control
- Claims Records
- Demand Deposit Accounting
- Commercial Loan Accounting
- Savings

Businesses, industries, and organizations that may benefit from the implementation of data base/data communication systems using CICS include:

- Business Services
- Distribution
- Education
- Financial
- Governments
- Insurance
- Manufacturing
- Media
- Medical
- Process
- Public Utilities
- Transportation

The data base/data communications system illustrated in Figure 4 graphically depicts the relationship of CICS to the user-written application programs, the operating system, the computing system, the data base, the real-time support, and the data communications.

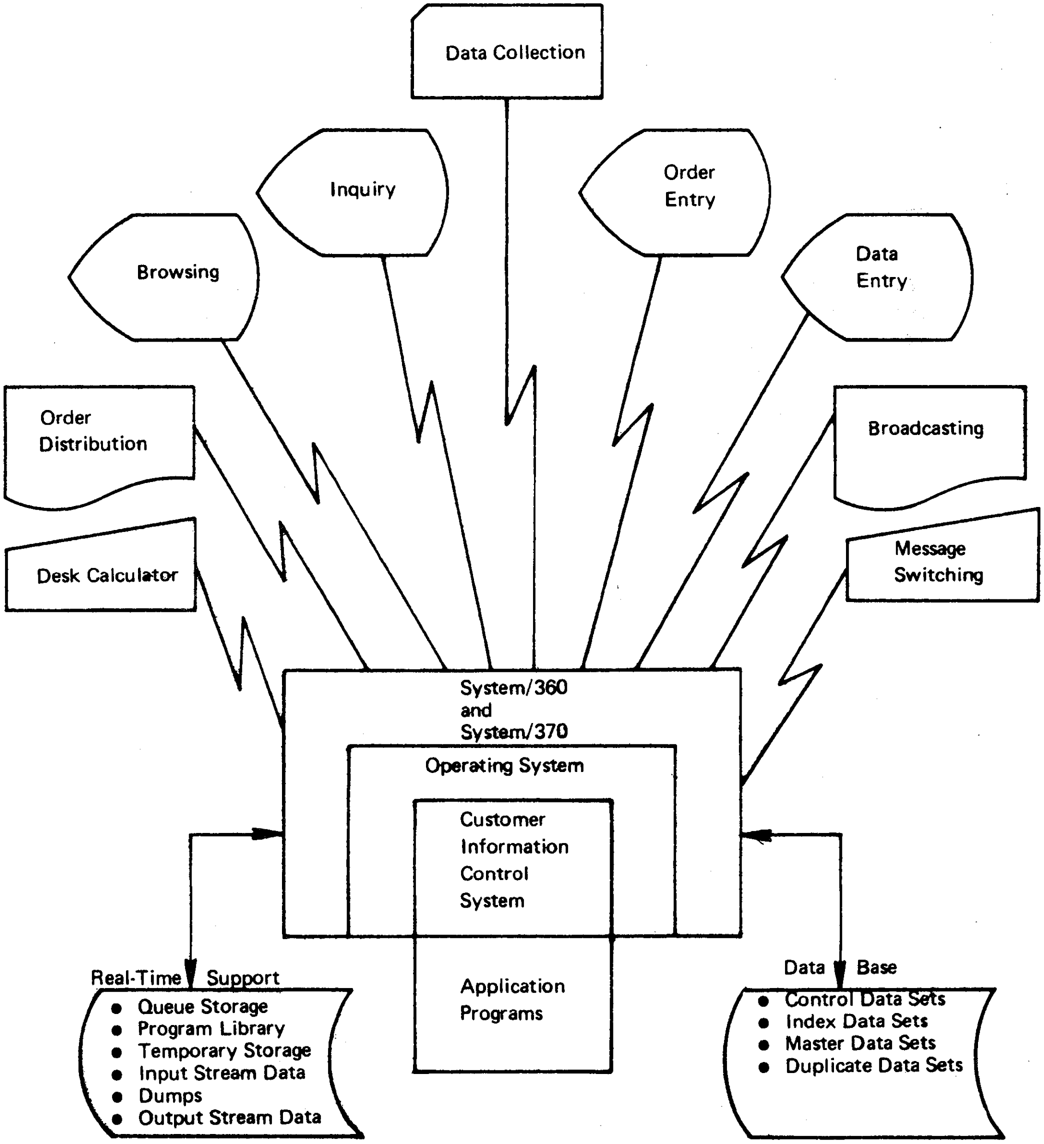


Figure 4. Data base/data communication system

INPUT/OUTPUT DESCRIPTION

Input/output data in CICS is, for the purpose of this description, divided into two somewhat arbitrary categories: system and user. This is a reasonable division even though, to some extent, both the user and the control system (CICS) are involved with all input/output.

SYSTEM INPUT/OUTPUT

System input/output is generally in support of the real-time environment. The access methods and record content of the system data sets are predefined within CICS. The system data sets (files) include:

- Real-Time Relocatable Program Library
- Real-time Pre-Located Program Library
- Transaction Rollout Data Set
- Dump Data Set
- Intrapartition Data Set
- Temporary Storage Data Set

REAL-TIME RELOCATABLE PROGRAM LIBRARY

The Real-Time Relocatable Program Library contains all user programs and CICS programs to be loaded and executed in real-time including the control system itself and certain user-prepared System Control Tables. The library contains program text and, where applicable, a relocation dictionary for each program. The contents of this library are asynchronously fetched into main storage for real-time execution in the CICS/DOS-STANDARD and CICS/OS-STANDARD systems by the Program Control program. This library is not used for real-time execution in the CICS/DOS-ENTRY system.

REAL-TIME PRE-LOCATED PROGRAM LIBRARY

The CICS Real-Time Pre-Located Program Library is used in the CICS/DOS-ENTRY system to retain the nonresident application programs during real-time execution. All application programs, so designated by the user, are loaded into main storage from the Real-Time Relocatable Program Library by the CICS System Initialization program for a given execution, and are written into the library for subsequent fetch and refresh. The Processing Program Table in main storage serves as the directory for this library.

TRANSACTION ROLLOUT DATA SET

The Transaction Rollout data set (file) is used in the CICS/DOS-ENTRY system to retain certain transaction information in order to support conversational interaction with a terminal. The types of transaction information that may be expected to be rolled into this data set include:

1. Task Control area
2. Transaction Work area
3. User-acquired work areas
4. File, Temporary Storage, Transient Data input/output areas, work areas, and control areas
5. Register storage areas
6. Loaded programs

DUMP DATA SET

The optional dump data set is used by the Dump Control program to record dumps of transactions within the system. It is a sequential data set located on either magnetic tape or direct access and can be subsequently formatted and printed by the CICS Dump Utility program.

CICS provides the capability to open/close the active dump data set during the real-time execution of the system. Optionally, the user can define two dump data sets (DFHDMPA and DFHDMPB), alternating between them during real-time execution of CICS.

INTRAPARTITION DATA SET

The Intrapartition data set is a system option and is used for the queuing of user data and, optionally, CICS data by the Transient Data Control program (CICS's general purpose queuing facility). Data, as requested, is stored chronologically into this data set according to previously identified symbolic destinations for subsequent retrieval when all applicable system conditions and facilities are in the appropriate status. This data set is required for applications such as message switching, broadcasting, order distribution, etc. It is also required if the CICS Asynchronous Transaction Processing Facility is being used. The record format for this data set is standard System/360 variable length.

TEMPORARY STORAGE DATA SET

The Temporary Storage data set is a system option which provides general purpose scratch pad storage for use by the Temporary Storage Control program. User data, as requested, is stored into this data set under a dynamically-provided symbolic identification for subsequent retrieval and release (when appropriate). This data set is intended for applications such as broadcasting, video display paging, transaction suspension, main storage conservation, etc. This data set consists of variable-length records within a preformatted fixed-length Direct Access Method (DAM) data set record.

USER INPUT/OUTPUT

User input/output is all input/output associated with the data base/data communication environment not described as system input/output, and includes:

- Terminal input/output
- The user's data base
- The extrapartition data sets

TERMINAL INPUT/OUTPUT

With the exception of invoking and interacting with the CICS service functions (application level processing programs), the user largely controls and prescribes both the format and content of his terminal input/output. Terminal input satisfactorily received is translated (as applicable) and passed, with the message length, to the user's application program for processing. Terminal output is prepared by the user's application programs and submitted to CICS for transmission to the desired terminal.

DATA BASE

CICS File Management allows the user a high degree of flexibility when defining the structure of his data base. Individual data sets (files) within the data base can be accessed under control of the Direct Access Method (DAM) or the Indexed Sequential Access Method (ISAM). Optional CICS features allow certain combinations of these data set organizations to provide the user with the most efficient data base structure suitable to his environment.

Segmented Records

With the segmented record capability, it is possible to retrieve an individual record or selected segments of an individual record. A segmented record is one in which the components of the record have been identified and grouped according to frequency of use, function, and logical relationship. The identifiable groups are called segments. Some segments, such as those that contain identification or major record control fields, would appear in all records. Other segments might appear only in certain records. The primary reason for segmenting records is to conserve main storage, or, in the case of variable length records, to conserve direct access storage.

If additional information is to be included in a segmented record, either a new segment is created or an existing segment is altered. In either case, only the affected segment descriptions need be changed and the new programming added to support the change. Symbolic references to unchanged segments are not affected. Adding information to a fixed format record could require considerably more programming than using the segmented record approach.

The user must define the record segments to CICS. In addition, each record to be retrieved in segments must have control information in the first segment which is used to indicate the presence or absence of each segment. A segment should contain logically related data so that only selected segments are required to satisfy the processing requirements of a transaction. A transaction that uses only selected record segments requires less main storage for its processing.

The user, in selecting those segments necessary for processing transaction groups, identifies them to CICS as a part of the data set (file) definitions in the File Control Table. Such a group of segments is a segment set. A segment set can include a single segment or all segments of a record. When a request is made to File Control identifying a segment set in the request, CICS always returns the header control segment plus the segments in that set.

Segmented records can be used with either DAM or ISAM data set organizations.

Deblocking Services for DAM Data Sets

CICS provides deblocking of logical records which are blocked and written out on a direct access (DAM) data set. This service is provided for both fixed- and variable-length records. The data set must be created according to standard System/360 record formatting conventions.

Indexed Data Sets - Indirect Accessing

CICS (optionally) allows the use of cross-index data sets to access another data set, which may be the main data set or another level of index data set. If a record retrieved from a cross-index data set indicates multiple entries in the main data set, information is returned to the user-written application programs to be used in selecting the appropriate main data set entry. When the cross index does not indicate multiple entries in the main data set, the File Control program reads the requested record from the main data set.

Organization of the cross-index data set may be either indexed sequential or basic direct access. The index record contains, in addition to the information used to find it, the search argument for the record on the data set which the index data set references. The index record may contain any other information desired by the user. The location of the search argument, its length, and the data set identification for the referenced data set are supplied to CICS as part of the data set definitions in the File Control Table.

DOS ISAM Variable-Length Records

The CICS/DOS systems support the retrieval and static update (no length variation) of variable-length records within fixed-length blocks under an ISAM organization. These pseudo-variable blocks must conform to System 360/370 variable-length record format conventions. That is, the first four bytes must contain the block length of the form LLbb. Since all blocks are fixed length, this value is the same for all blocks. Each logical record within the block must reflect the length of the record in the first four bytes (LLbb). A logical record may not be continued onto the next block. The first byte of any unused portion of a block must contain a hexadecimal FF.

The addition and deletion of records on a DOS ISAM variable-length record data set must be handled by the user in an offline batch environment. When creating the data set, it must be defined as fixed unblocked, and the key for each block must be the same as the last logical record in that block. The block size must be an even number of bytes. All records must reside in the prime data area; no overflow records are allowed.

Data Language/I (DL/I)

The CICS/OS-STANDARD system provides optional access to the Data Language/I (DL/I) facility of the IBM Information Management System Version 2 (IMS/360). Use of DL/I requires installation of the IMS/360 Version 2 Data Base System (5734-XX6) and generation of the required control block libraries.

The DL/I Program Specification Blocks (PSBs) are generated in the regular IMS fashion, but the application programs that use the PSB's are placed in the CICS load library and are defined in the Program Control Table and the Processing Program Table. The main storage requirement for a CICS system that uses DL/I is the sum of the requirements for both the CICS system and the IMS Data Base system.

Support for DL/I is provided through an interface between the CICS File Control program (FCP) and a DL/I batch program executing as an OS subtask of CICS. The interface processes one DL/I CALL at a time, allowing several transactions that request DL/I services to be active simultaneously. If a transaction issues a "get for update" request ('GHxx' CALL) against a data base, all other get for update requests from other transactions against that data base are queued (stored) until the first transaction either issues some other type of request (CALL) or that transaction terminates. In this manner, concurrent update of a data base by two transactions is prevented. However, other transactions may issue inquiry-only calls against the data base while a transaction is updating that data base.

Access to DL/I data bases is through either of two means: (1) DL/I CALL's formulated according to DL/I specifications, or (2) via the CICS File Control macro instruction (DFHFC) using unique DL/I operands. In either case, the request is processed by CICS routines and then by the DL/I facility.

For complete information on the capabilities of DL/I, the formulation of DL/I CALL's, and the generation of an IMS Data Base system, refer to the appropriate IMS/360 Version 2 manuals as listed in the "Bibliography".

EXTRAPARTITION DATA SET

Extrapartition data is the name given in CICS to transient stream data which is coming into or going out of the data base/data communication system environment. This data is usually coming from or going to high-speed magnetic devices and is characterized by, but not required to be, blocked variable-length stream data. The extrapartition disposition facility is intended for use with the following types of data and applications:

1. Message logging
2. Transaction logging
3. Reconstruction records and information
4. Data collection
5. Data entry

and all other output stream data intended for subsequent processing (usually offline). The user defines this data consistent with his requirements and facilities. The extrapartition acquisition facility is intended for limited stream (batch-like) data processing in the data base/data communication environment.

PROCESSING DESCRIPTION

For purposes of description, CICS can be organized into three major divisions:

- Real-Time Management Functions
- Real-Time System Service Functions
- Components That Are Not Real-Time

Each of these divisions can be further divided into functional subdivisions to provide the appropriate level of description.

The same modular design of CICS which allows this descriptive subdivision also allows the user to tailor and configure a system appropriate to his individual requirements. While some of the components are essential to the system, many are optional and may be configured into the system under user control.

Although portions of the logic of some of the components may differ from one CICS system to another, the function, service, and interface are the same, thereby making any architectural differences transparent to the user. At this level of discussion, any such differences are also transparent.

Figures 5, 6, and 7 graphically illustrate this point. Only minor differences can be detected between the three systems. It is this same transparency in every aspect of the system design that facilitates user migration as his requirements grow, without having to change his system design or program code.

Also shown in Figures 5, 6, and 7 are the relationships between the physical environment (terminals, data sets, etc.), the operating system (OS, DOS, etc.), the CICS management functions, and the application programs.

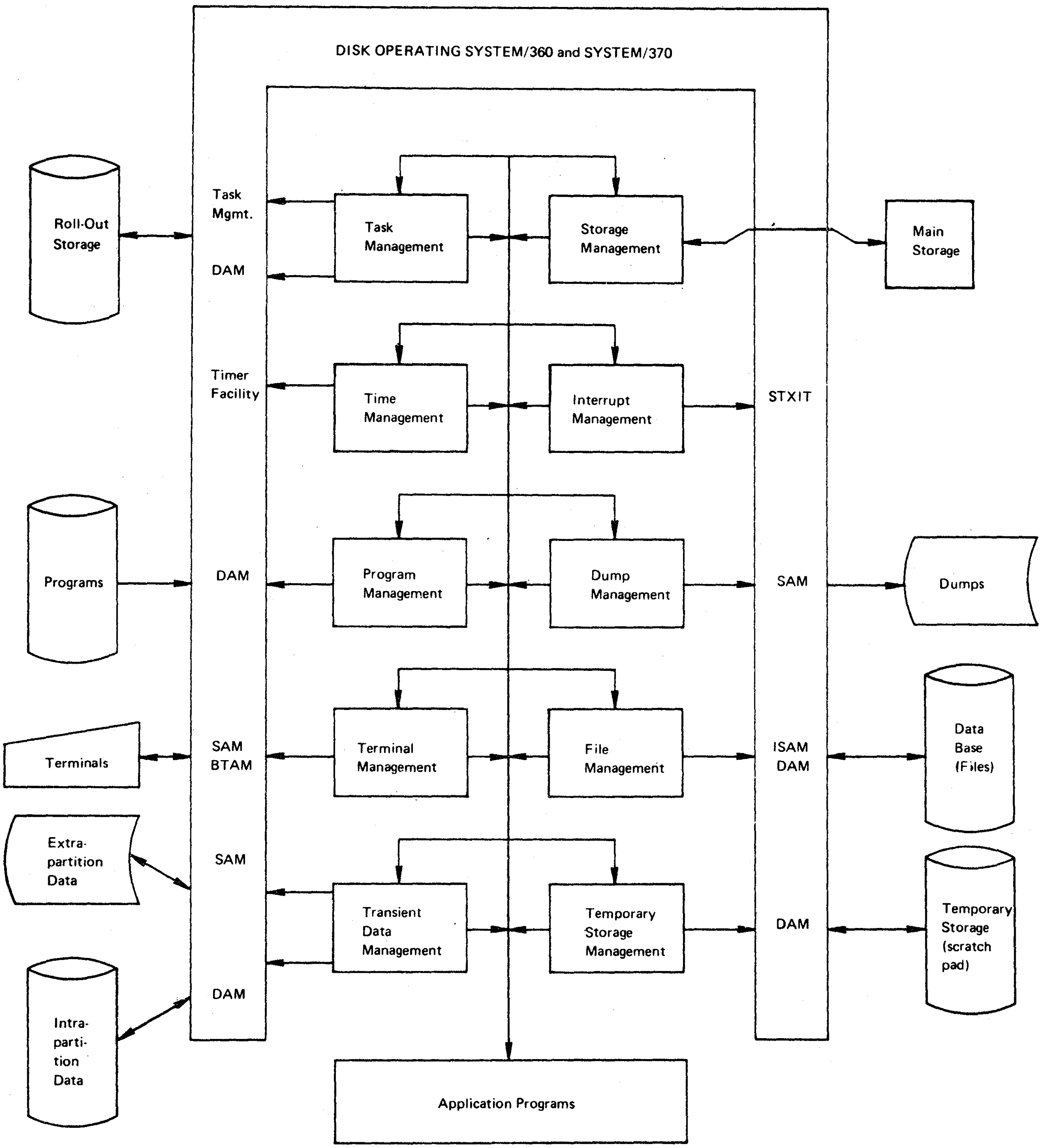


Figure 5. CICS/DOS-ENTRY system

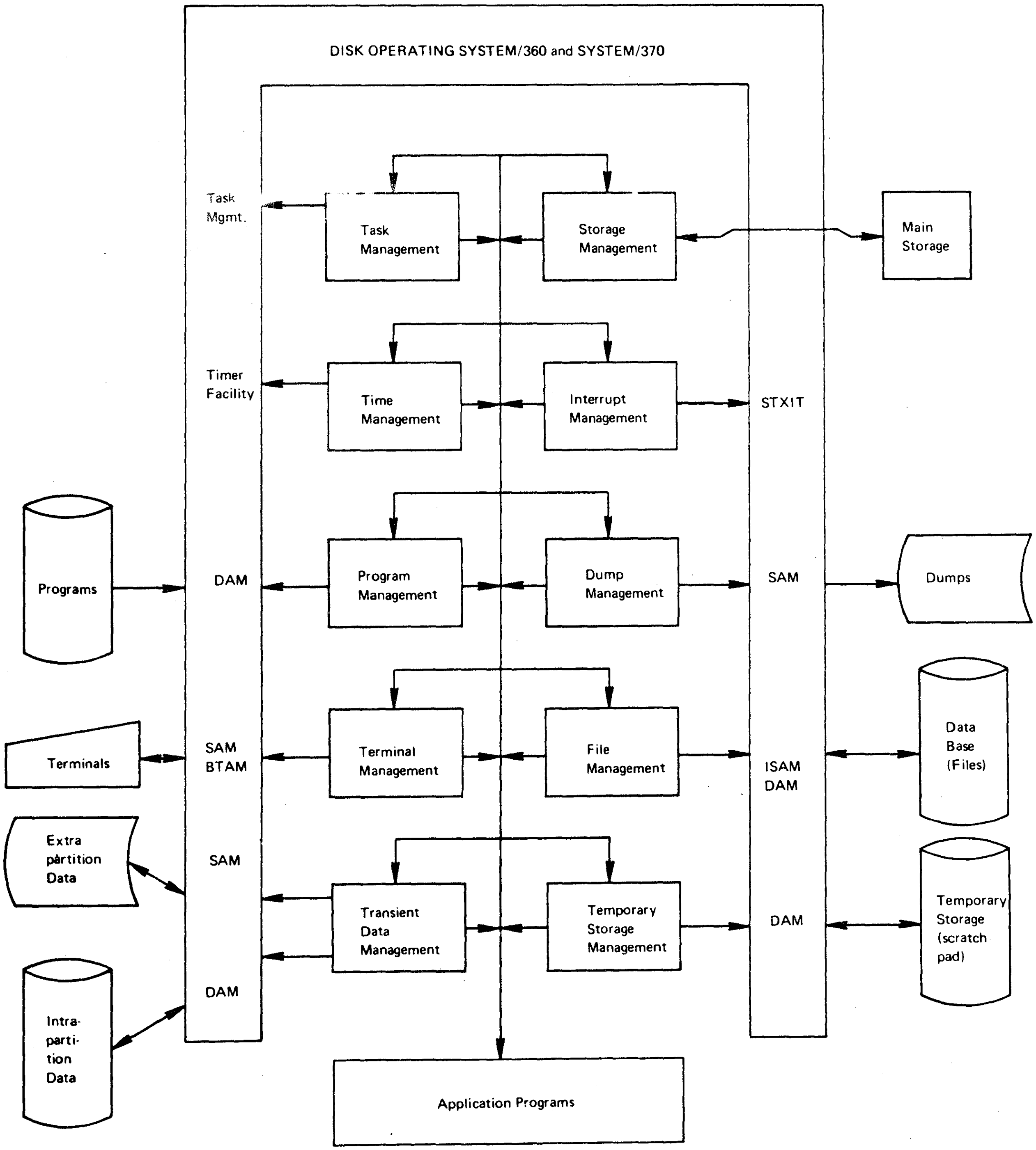


Figure 6. CICS/DOS-STANDARD system

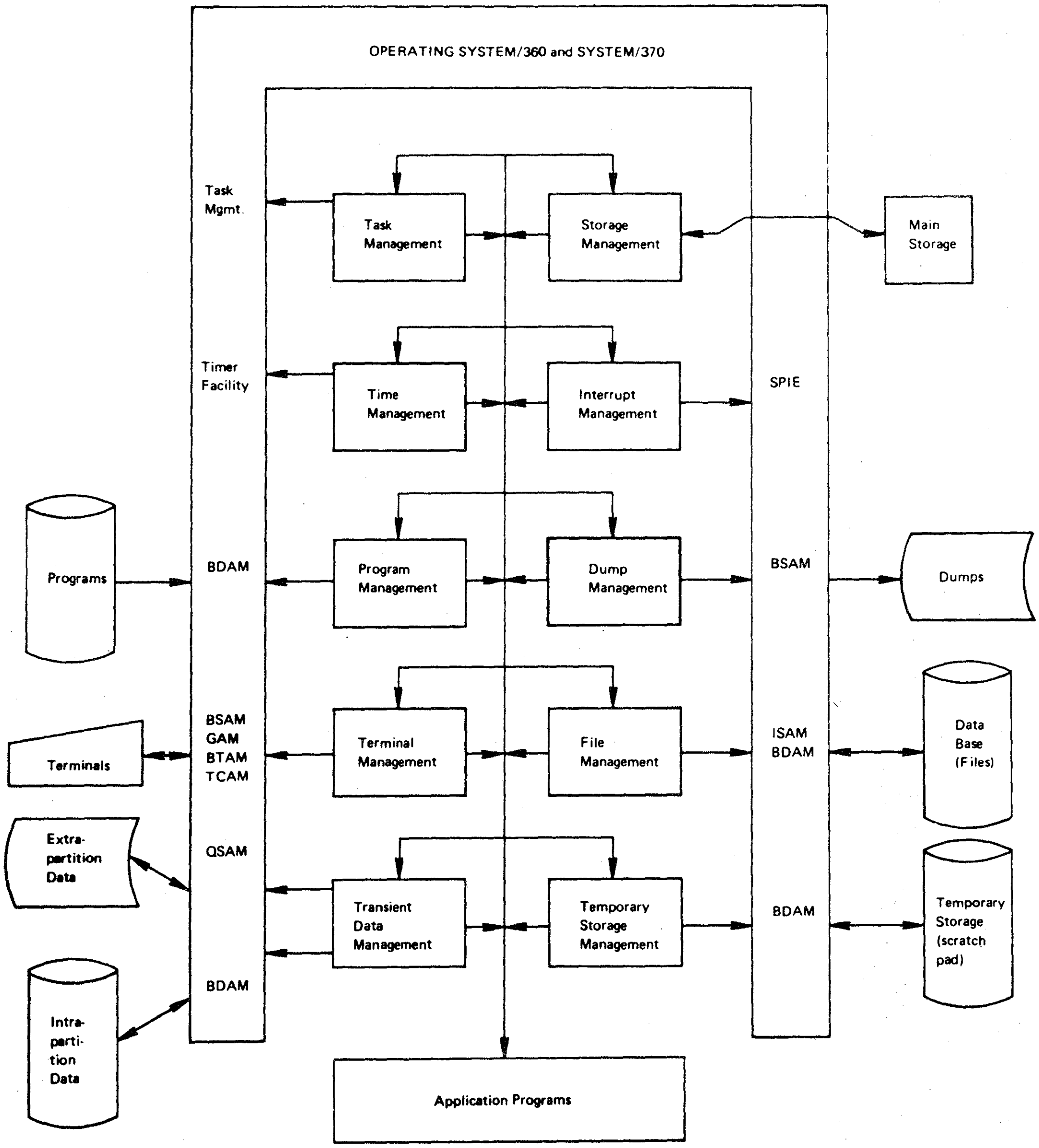


Figure 7. CICS/OS-STANDARD system

REAL-TIME MANAGEMENT FUNCTIONS

The real-time management functions, as with any operating system, may be best understood if divided according to their associated services into the categories of supervisory and data management functions.

Supervisory functions include:

- Task Management
- Storage Management
- Program Management
- Program Interrupt Management
- Time Management
- Dump Management

Data management functions include:

- Terminal Management
- File Management
- Transient Data Management
- Temporary Storage Management

TASK MANAGEMENT

Transactions are logical system functions that are composed of one or more inputs and outputs. The ability to process multiple transactions concurrently is provided by Task Management through the Task Control program. Transactions are scheduled and processed according to priorities assigned by the user. Control of the central processing unit (CPU) is given to the highest priority task that is ready. Control of the CPU is returned to the operating system when no further work can be done by the user-written application programs.

Communication between the Task Control program and user-written application programs is via CICS macro instructions. Task Control:

1. Initiates (originates) a valid task (a unit of work for the CPU). When a message is read by the Terminal Control program, the Terminal Contrcl program requests that a task be initiated to process the message.
2. Terminates a task. When the task is complete, notification is given to Task Control to remove the task from CICS control.
3. Changes the priority of a task.
4. Delays the dispatch of a task until an awaited event is complete. Each task gives up control of the CPU when the task makes a request for a CICS service that requires an input/output operation. Task Control does not allow the task to resume processing until the input/output event is complete.
5. Queues requests for use of a serially reusable resource which is currently being used to service a prior request.

The number of tasks that are active is limited by the amount of available main storage and/or by the number of tasks specified as a maximum by the user. No new tasks are created unless there is sufficient main storage for processing the tasks. The user has the option of specifying that only a certain number of transactions be processed concurrently; if the user-supplied limit is included, no task is created that causes the limit to be exceeded. The task is enqueued and subsequently created at a time when the limit is not exceeded.

Statistics maintained by the Task Control program include the maximum number of tasks within the system at a given time and the total number of tasks created in a given time period.

Optional useт exits are provided as follows:

1. In the Task Dispatcher after determination of which task to dispatch.
2. Prior to determining what type of request for task services was issued.

Task Control Functions

The following functions are provided automatically by Task Management without any communication from the application program.

TASK DISPATCHING: Passes control from Task Control to the highest priority task in the active task queue that is ready to use the central processing unit.

MAXIMUM TASK CONTROL: Indicates when the user-specified maximum number of tasks within the system has been reached.

STALL DETECTION AND CORRECTION: Detects the system stall condition where main storage resources available to CICS become overloaded to the point where active transactions cannot continue processing and new transactions cannot be initiated. Corrective action involves the purging of low-priority "purgeable" transactions designated as purgeable by the user.

RUNAWAY TASK DETECTION AND CORRECTION: Detects the condition where an application program may have developed a "loop" within the program logic. Corrective action involves the abnormal termination of the task (transaction).

The following functions are provided by Task Management in response to macro instructions issued in the application program.

TASK ORIGINATION: Creates the controlling data for a task and places the task in the active task queue.

TASK TERMINATION: Deletes the controlling data for a task and removes the task from the active task queue.

TASK SYNCHRONIZATION: Synchronizes a task with the completion of an event that is in another task or that is part of a CICS service. For example, for a task that requests a GET operation (a CICS service), processing does not resume until the GET is complete; therefore, other tasks may be processed while this task is waiting.

PRIORITY CHANGE: Changes the task dispatching priority for an existing task and rearranges it in the active task queue.

RESOURCE SYNCHRONIZATION: Controls the use of serially reusable resources, based on user-defined symbolic conventions.

TASK SUSPEND: Removes the addressed task from the active task queue and places it on the suspended task queue to optimize active task dispatching.

TASK RESUME: Removes the addressed task from the suspended task queue and places it on the active task queue to be executed in turn.

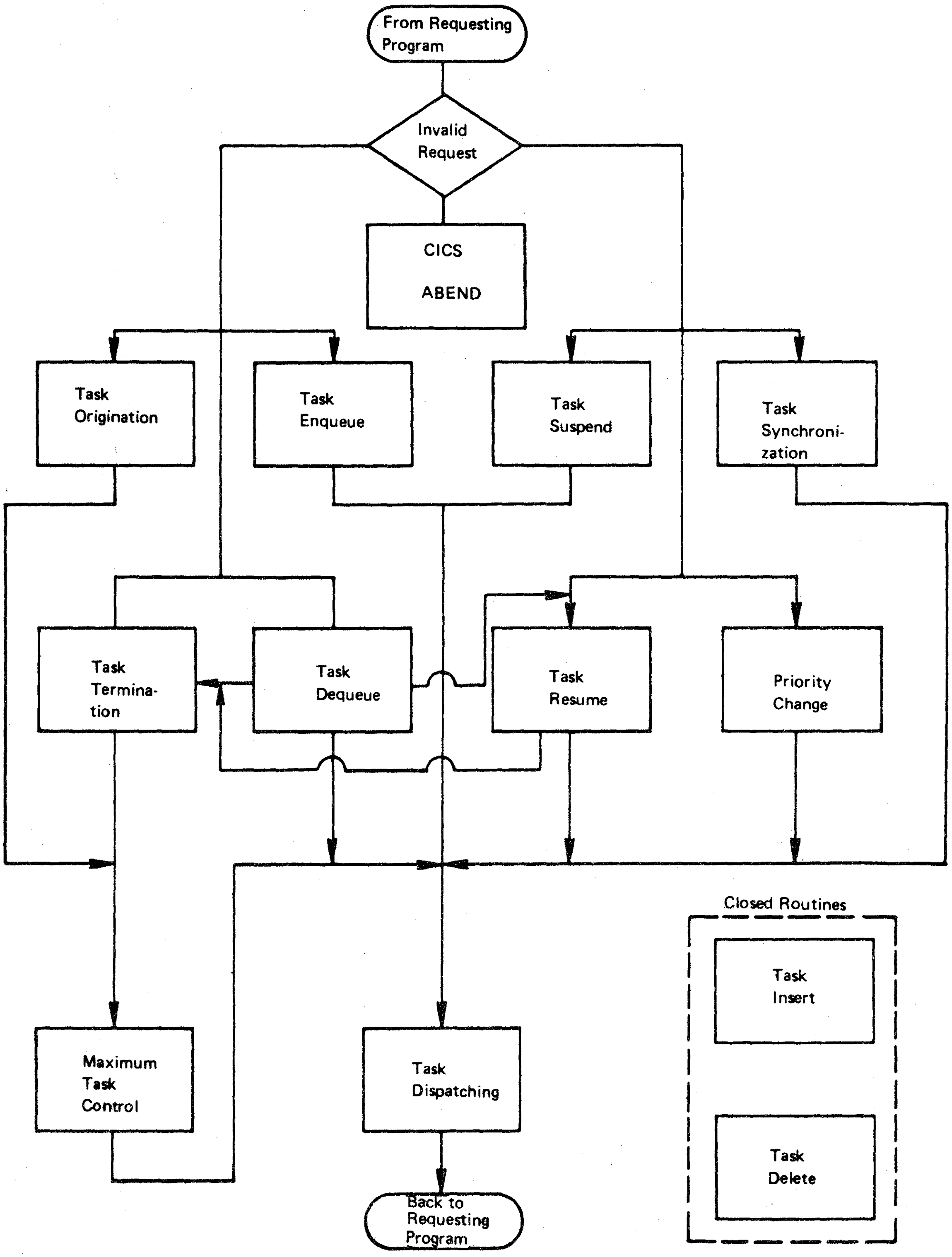


Figure 8. Task control program

STORAGE MANAGEMENT

All main storage for CICS and for the user-written application programs is controlled by Storage Management through the Storage Control program. Requests to acquire or release main storage are communicated to Storage Control by CICS macro instructions. CICS requests for main storage are for input/output areas, program load areas, and user-defined work areas that are to be supplied for a transaction. User-written application program requests for main storage are for intermediate work areas and for transaction processing. Acquired main storage is initialized to any bit configuration specified as an option within the requesting macro instruction. For example, a main storage area could be initialized to all binary zeros or EBCDIC blanks.

All storage areas used for a transaction are chained. This feature allows CICS to release all main storage associated with a transaction either upon request by the user or when the transaction is abnormally terminated.

The user can issue either a conditional or unconditional request to acquire main storage for a transaction. When there is insufficient main storage to satisfy a storage acquisition request, a conditional request allows control to be returned to the user with an indication that there was insufficient main storage; an unconditional request results in the transaction being suspended until sufficient storage is available. No new tasks are initiated by CICS until the "short on storage" condition is alleviated. The only exception to this method of allocating main storage occurs in the CICS/DOS-ENTRY system where, under certain circumstances, a "short on storage" condition causes the transaction to be abnormally terminated unless the storage acquisition request was conditional. In this manner, the Storage Control program modulates the effects of high storage demands and keeps the system in operation.

To aid in this process, CICS uses a technique which involves a "storage cushion." The cushion is acquired during System Initialization and is held in reserve by CICS until all available storage is in use. Any additional storage requests cause the cushion to be released and initiation of new tasks to be inhibited. When the demands for main storage resources have diminished, the cushion is reacquired by CICS and new transactions are again initiated.

Statistics maintained by the Storage Control program include the number of requests for storage acquisition, the number of times processing must be delayed for storage acquisition, and the number of requests for storage disposition.

An optional user exit is provided during entry analysis.

Storage Control Functions

The following functions are provided automatically by Storage Management without any communication from the application program.

STORAGE REQUEST ENQUEUING AND DEQUEUING: Enqueues and dequeues requests for main storage which cannot be immediately satisfied.

SYSTEM NOTIFICATION: Inhibits the initiation of new work. This is an internal function; no external notification is given.

STORAGE ACCOUNTING: Automatically chains storage acquired by a transaction and releases any unreleased storage upon termination of the transaction.

The following functions are performed by Storage Management in response to CICS macro instructions issued in user-written application programs.

STORAGE ACQUISITION: Allocates dynamic main storage for a transaction upon request.

STORAGE DISPOSITION: Releases main storage for a transaction upon request.

STORAGE INITIALIZATION: Initializes allocated main storage to the bit configuration specified by the user in the macro instruction.

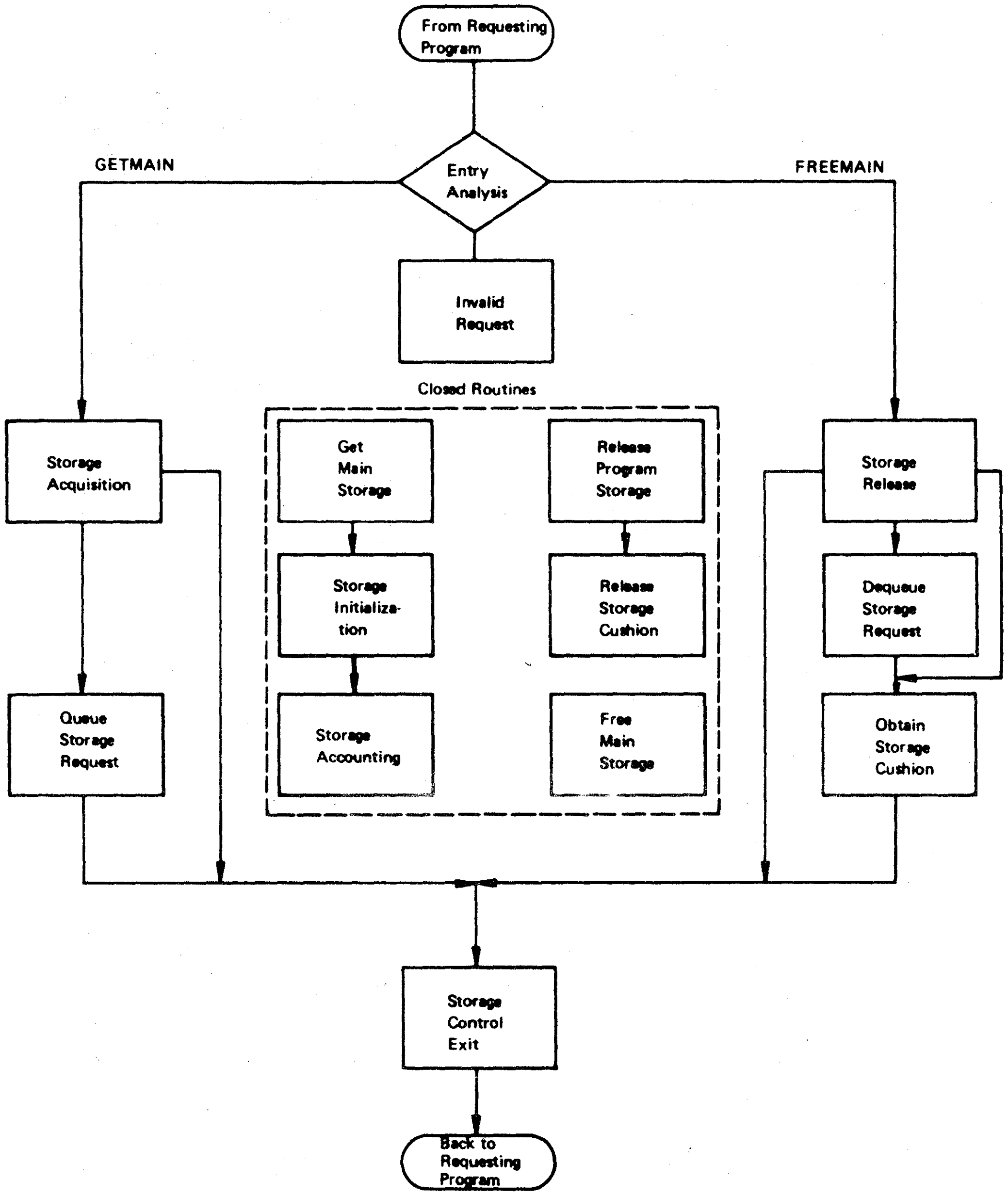


Figure 9. Storage control program

PROGRAM MANAGEMENT

Program Management functions within CICS are controlled by the Program Control program. Program Control macro instructions are used to load programs, link or transfer control to programs, delete a loaded program, abnormally terminate a program, and return control from a program. The normal termination of tasks and the termination of transaction processing are additional services initiated by this program. Single copies of user-written programs in main storage are controlled to allow concurrent use by multiple tasks.

User-written application programs must be written so that they are serially reusable between the entry and exit points of the program. Entry and exit points of an application program coincide with the use of CICS macro instructions, since an application program temporarily loses control after it begins executing only upon execution of a CICS macro instruction. A serially reusable portion of an application program is executed by only one transaction at a time and must initialize and/or restore any instructions or data that it alters within itself during execution.

This required quality of application programs to run under CICS is called "quasi-reentrance", since the programs need not meet System/360 or System/370 requirements for true reentrance. Quasi-reentrance allows a single copy of a user-written application program to be used to process several transactions concurrently, thereby reducing the requirement for multiple copies of the same program in main storage. The user may use common work areas between entries and exits, and is required to use unique storage areas only for those data items that must be retained when the user-written program exits to CICS. Programs written to run under CICS/DOS need not be self-relocating.

When a requested user-written application program is already in main storage, the Program Control program transfers control directly to that program. Each program's location on a direct access volume and in main storage, if applicable, is kept in a Processing Program Table (PPT). The status of all programs is maintained in the PPT and reflects whether a program is in main storage and whether it is in use.

Programs are stored on a CICS library in relocatable format and are accessed through the CICS asynchronous program fetch data management facilities. This loading facility allows program execution to continue during program load time. For CICS/DOS-STANDARD, the Real-Time Relocatable Program Library is a user-allocated extent. Programs are prepared for this library by DFHLINK. For CICS/OS, the Real-Time Relocatable Program Library is a standard partitioned data set. Programs are prepared for this library by the OS Linkage Editor. For CICS/OS, users may concatenate other private libraries to this library.

As control for a task is linked from one processing program to another and returned, Program Control saves and restores the general purpose registers. When control is returned to Program Control at the completion of task (transaction) processing, Program Control issues a request to delete the task which, in turn, releases all storage associated with the task. The areas that are released include the Transaction Work Area, the Task Control Area, input/output areas, and general storage areas that were obtained by the processing programs.

Application programs remain resident in main storage when not being used unless there is an indication that system storage resources have become overloaded. In this case, programs not currently in use are deleted from main storage.

Application programs can be written in Assembler language, ANS COBOL, or PL/I to execute under CICS. Regardless of the language used, it is strongly recommended that the application programmer allow CICS to perform all supervisory and data management services for his applications by issuing the appropriate CICS macro instructions. Although the application programmer is not precluded from direct communication with the operating system, the results of such action are unpredictable and performance may be affected. It would also have a limiting effect on migration from CICS/DOS to CICS/OS, if this were ever desired.

An optional user exit is provided from Program Control after program fetch.

Program Management Functions

The following functions are automatically performed by Program Management without any communication with the application program.

HIGH-LEVEL LANGUAGE (HLL) INTERFACE: Intercepts HLL program requests for CICS services to save information, passes control to the appropriate CICS program, and subsequently returns to the HLL program.

PROGRAM PURGE: Causes unused application programs to be purged from main storage when CICS is in the overload state. Normally, all application programs remain resident in main storage, even when not in use.

ASYNCHRONOUS PROGRAM FETCH: Provides for the fetching of application programs from the direct access storage device into main storage while allowing other transaction activity to proceed during the I/O operation.

The following functions are performed by Program Management in response to macro instruction issued in the application program.

LINK: Retain requesting program and its general registers for subsequent return, and pass control to the program specified in the macro instruction. The linked-to program is considered to be at the next lower logical level.

TRANSFER CONTROL: Transfer control from one user-written application program to another. The transferred-to program is considered to be at the same logical level.

LOAD: Load the designated program into main storage and return its entry address to the requesting program.

DELETE: Release the designated program which was previously loaded.

RETURN: Return control to the program considered to be at the next higher logical level. This leads to normal transaction termination.

ABEND: Terminate a transaction and its related task and pass control to the Abnormal Condition program.

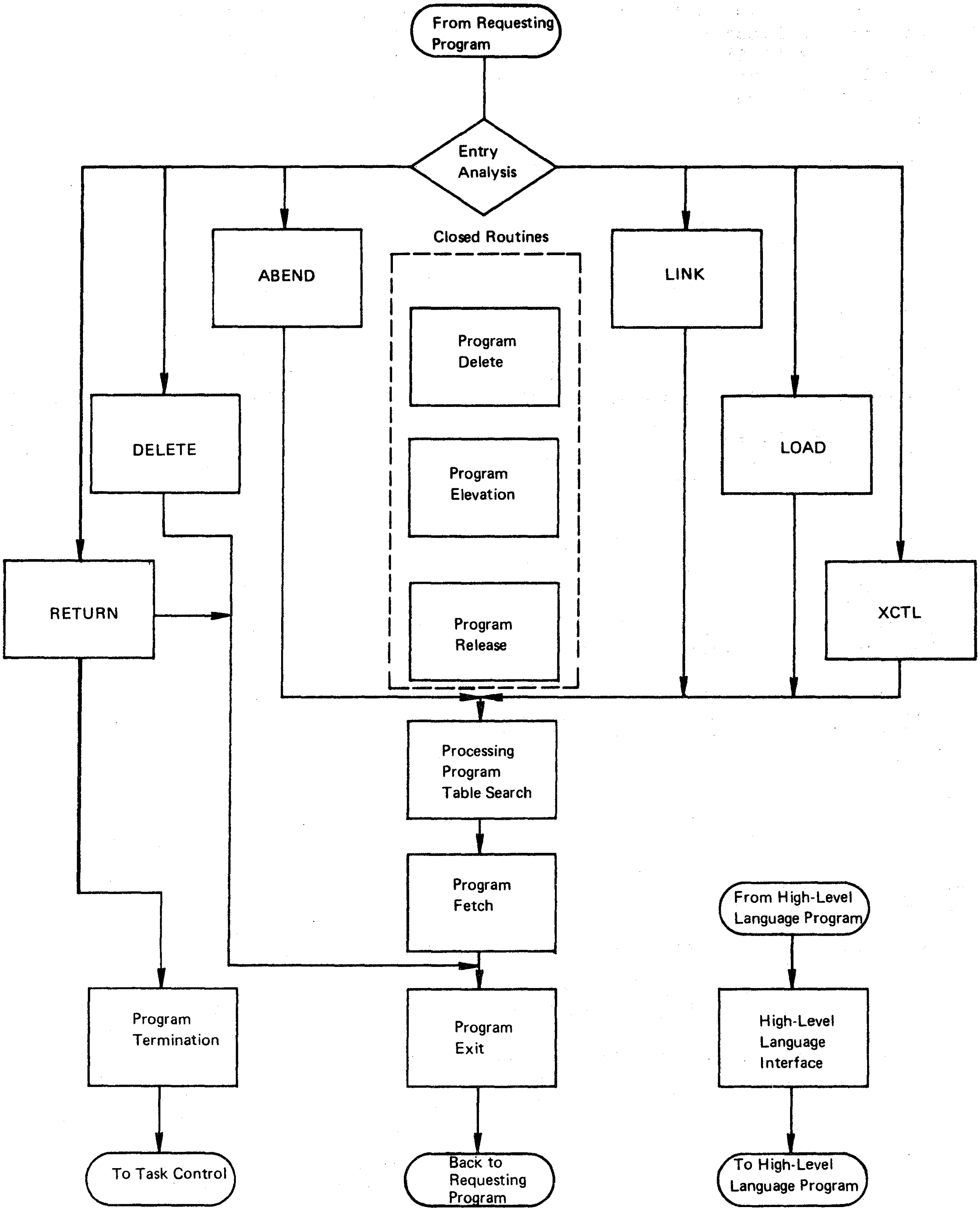


Figure 10. Program control program

PROGRAM INTERRUPT MANAGEMENT

The Program Interrupt program is a generalized program interrupt handler which is given control by the operating system when a program interrupt occurs within the CICS partition/region. Its use is optional, and may or may not be included at System Initialization. If included, System Initialization establishes the necessary operating system linkage by issuing the STXIT (DOS) or SPIE (OS) macro instruction.

If a program check occurs, the transaction which has control of CICS is abnormally terminated with a CICS dump (if Dump Management is included); other transactions are allowed to continue processing.

TIME MANAGEMENT

Certain system functions and services are available to the user through the optional Time Management facility. Time Management functions are controlled by the Task Control program and the Interval Control program.

Time Management Functions

The following functions are performed by CICS, based on intervals of time specified by the user during System Initialization. They require no communication with user-written application programs.

PARTITION/REGION EXIT TIME INTERVAL CONTROL: The partition/region exit time interval is the maximum interval of time that CICS will release control to the operating system in the event there are no transactions ready to resume processing.

SYSTEM STALL DETECTION AND CORRECTION: Provides automatic detection of the system stall condition where CICS main storage resources have become overloaded to the point where no active transactions can continue and no new transactions can be initiated. Corrective action involves the purging of low-priority "purgeable" transactions designated as purgeable by the user. The status of all transactions is defined and controlled by the user.

RUNAWAY TASK DETECTION AND CORRECTION: Automatic detection of the situation where an application program may have developed a "loop" within the program logic. Corrective action involves the abnormal termination of the transaction.

The following functions are performed by Time Management in response to CICS macro instructions issued by a user-written application program.

TIME OF DAY: Provides the capability of retrieving the current time of day in either binary or packed decimal format.

TIME-DEPENDENT TRANSACTION SYNCHRONIZATION: Provides the user with three optional services:

1. WAIT, permits a transaction to temporarily suspend itself for a given period of time. When the time has elapsed, the transaction continues to execute.
2. POST, provides the means for a transaction to be notified when the specified interval of time has elapsed or the specified time of day occurs. The transaction proceeds to execute while the time interval is elapsing.
3. CANCEL, allows a transaction to terminate its own or another transaction's request for a WAIT or POST service.

AUTOMATIC TIME-ORDERED TRANSACTION INITIATION: Provides for the automatic initiation of a transaction at a specified time of day (or after a specified interval of time has elapsed) and for the control of data which is to be accessed by the transaction. The capability to cancel a pending request for automatic time-ordered transaction initiation is also provided.

Optional user exits are provided as follows:

1. Prior to determining what type of request for time services was issued.
2. Upon expiration of a previously requested time-dependent event.

DUMP MANAGEMENT

Dump Management provides a diagnostic facility to assist in analysis of programs and transactions undergoing development or modification. This facility locates and outputs main storage images of specified CICS storage areas and user storage areas for subsequent formatting. Requests for storage dumps are communicated to the Dump Control program through CICS macro instructions; these dumps can include system control tables, storage areas, and programs related to a transaction. The areas processed by the Dump Control program are placed in a sequential data set for subsequent printing by the CICS Dump Utility program.

The Dump Control program records dumps of transactions on sequential data sets located on either magnetic tape or direct access. CICS provides the capability to open/close the active dump data set during real-time execution of the system.

Optionally, the user can define two dump data sets (DFHDMPA and DFHDMPB), alternating between them during real-time execution of CICS. This is accomplished in essentially a "flip-flop" manner; that is, if DFHDMPA is opened by the System Initialization program and if a request is issued to use the alternate dump data set, DFHDMPA is closed and DFHDMPB is opened. Another request to use the alternate dump data set causes DFHDMPB to be closed and DFHDMPA to be opened.

When a storage dump is in progress, the Dump Control program delays the processing of subsequent dump requests until the dump is completed.

Dump Control Functions

DUMP: Dumps system control tables, storage areas, and programs as requested.

ENQUEUE AND DEQUEUE: Enqueues and dequeues storage dump requests.

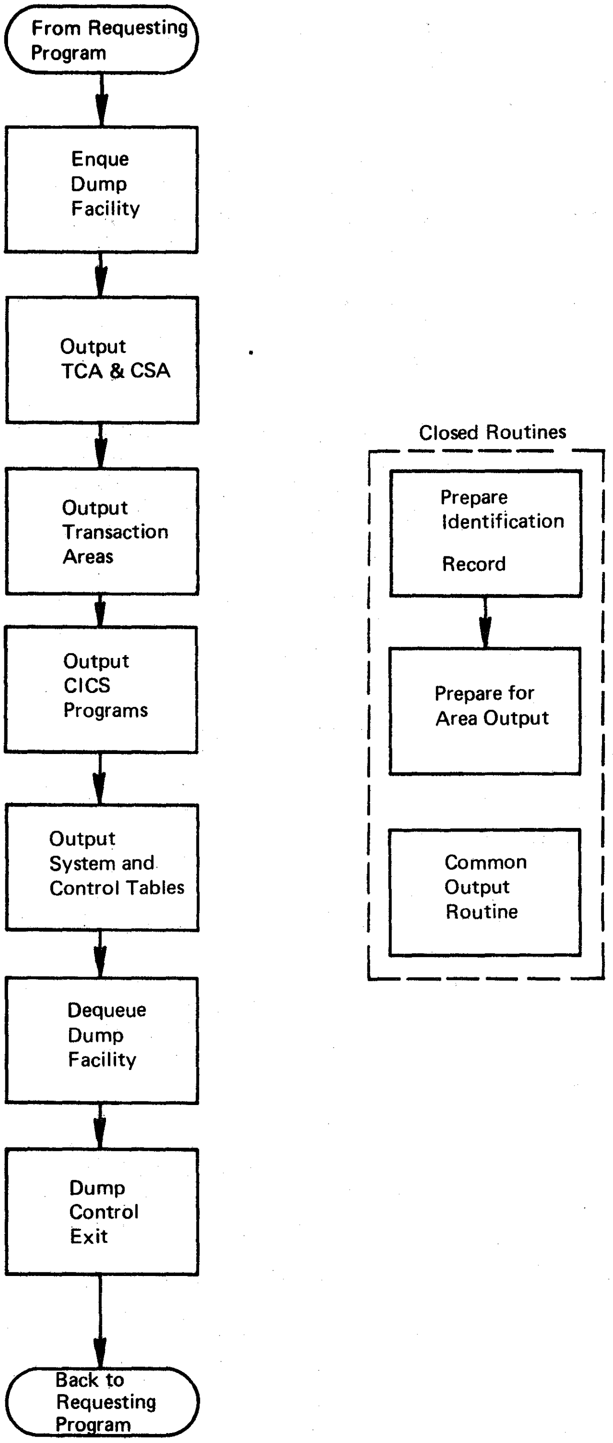


Figure 11. Dump control program

TERMINAL MANAGEMENT

Terminal Management provides for communication between terminals and user-written application programs through the Terminal Control program. The user selects only the CICS device-dependent submodules which pertain to the terminals on his system. Each of the submodules contains the linkage to an access method.

The Basic Telecommunications Access Method (BTAM) is used for most terminal data management. The Graphics Access Method (GAM) is used for the support of locally attached 2260's in the CICS/OS-STANDARD system. The Telecommunications Access Method (TCAM) can be optionally chosen where consolidation of functions is a system requirement or where sharing of devices is desired in the CICS/OS-STANDARD system. See the section "System Configuration" for supported devices and features.

To provide a testing facility for the user's programs, the Sequential Access Method (SAM) is used to control sequential devices such as card readers and printers, magnetic tape, and direct access storage devices. These sequential devices can be used to simulate terminal activity.

The Terminal Control program uses data that describes the communication lines and terminals. This data is kept in the Terminal Control Table and is generated from user-prepared control cards. The table entries contain terminal request indicators, status, statistics, identification, and addresses of I/O and related areas.

Reading data from terminals is a function of Terminal Control. The user specifies the terminal device characteristics and desired polling interval so that CICS is generated to satisfy his requirements. (A polling interval is the period of time between one attempt to read data from a terminal or group of terminals and the next attempt to read data from the same terminal or terminals.)

When a read is completed, the input data is converted to the Extended Binary Coded Decimal Interchange Code (EBCDIC). When there is need for a task to process a message, a task origination macro instruction is issued by Terminal Control. When data is to be transmitted to a terminal, processing programs execute a CICS Terminal Control write macro instruction. The translation of output data from EBCDIC to the appropriate terminal code is performed if required.

The communication serviceability facilities, which are an optional feature of BTAM, are used in CICS to increase data processing system availability by providing statistics and diagnostic aids for use in both repair and preventive maintenance. If TCAM is chosen by the CICS/OS user, the facilities to provide statistics and diagnostic aids for use in repair and preventive maintenance are available within TCAM.

The functions of Terminal Control are categorized as either transmission facility control functions (those functions which are normally related to the control of the communication lines) or terminal device-dependent control functions (those terminal functions which are dependent upon device type and access method). Figure 12 shows the functional flow of the Terminal Control program.

When TCAM controls communication lines under CICS/OS, those lines are no longer dedicated to the CICS partition/region. The user thus has the capability for a single terminal to access multiple programs in separate partitions/regions which are supported by TCAM. TCAM facilities available within the partition/region supported by TCAM include message switching, broadcasting, disk queuing, checkpoint/restart of the communication network, and TCAM terminal support.

Transmission Facility Control Functions

The following functions are performed by, or in conjunction with, Terminal Management in response to CICS macro instructions issued in user-written application programs.

GET TERMINAL STORAGE: Obtains storage for polling activity and for requested read operations.

FRFE TERMINAL STORAGE: Releases unused terminal storage after the completion of output events.

DETERMINE INPUT DATA LENGTH: Computes the length of input data.

EVENT TERMINATION: Provides a "reset poll" for certain terminal devices to service write requests.

I/O ERROR TEST: Provides I/O error detection. See the discussion of "Terminal Error Recovery" below.

AUTOMATIC TASK INITIATION: Services requests for automatic task (transaction) initiation from the Transient Data intrapartition data set.

TASK INITIATION: Requests the initiation of a transaction upon request from a terminal. When an input message is accepted, a task is created to do the processing.

TRANSLATION: Translates received data from transmission code to EBCDIC and data to be sent from EBCDIC to the correct transmission code.

LINE ADVANCE: Scans the Terminal Control Table to make line control information available for analysis.

LINE ANALYSIS: Analyzes the Terminal Control Table line control information to determine which terminal facilities require further action. For example, an indication that a communication line is free could indicate that a polling operation should begin.

ACCESS METHOD SELECTION: Passes control to the appropriate access method-dependent submodule based on the access method specified in the DFHTCT TYPE=LINE macro instruction. For example, when a communication line on which there are 2740 Communication Terminals is found to require action, control is transferred to a Terminal Control submodule which contains the BTAM control logic.

WAIT: Synchronizes the Terminal Control task with all other tasks in the system. When all terminal read and write operations are begun that can be started, and Terminal Control processing is complete, control is returned to the Task Control program to allow dispatching of other tasks.

For purposes of data handling, transactions are considered to be either synchronous or asynchronous. Synchronous transactions entered from a terminal are handled directly by the user's application program. Asynchronous transactions entered from a terminal are handled first by a CICS- or user-provided Asynchronous Transaction Input Processing program, which places them on a direct access queue as a batch. When all data has been received for a batch, the appropriate user-written application program(s) are initiated to process the data asynchronously with other activity on the terminal. Any output to be sent back to the terminal by the application is placed on a direct access queue and later written to the terminal by a CICS- or user-provided Asynchronous Transaction Output Processing program.

The Asynchronous Transaction Input Processor, Asynchronous Transaction Output Processor, and supporting functions are optionally generated as the Asynchronous Transaction Processing facility.

Optional user exits are provided:

1. Just prior to initiation of a transaction.
2. Just prior to starting an output event.
3. Following completion of an input event.

Terminal Device-Dependent Control Functions

For terminal device-dependent control, there is a module for each access method associated with the device type supported by CICS. Figure 12 shows the functional flow that is common to all terminal device-dependent modules.

INPUT EVENT TREATMENT: Processes a completed input event, including error checking, storage management, translation, and task initiation.

OUTPUT EVENT TREATMENT: Processes a completed output event, including error checking and storage management.

ACTIVITY CONTROL: Examines the control information for each terminal, checking for requested writes, reads, waits, and other Terminal Control macro requests.

INPUT EVENT PREPARATION: Prepares the line for an input event, including storage management.

OUTPUT EVENT PREPARATION: Prepares the line for an output event, including translation.

EVENT INITIATION: Prepares the terminal data event control block and prepares the linkage of terminal device-dependent control to the appropriate access method.

Terminal Error Recovery

The resolution of permanent transmission errors involves both CICS and additional user coding. CICS cannot arbitrarily take all action with regard to these errors. User application logic may be necessary to properly resolve the problem.

The following sequence of events takes place when a permanent error occurs for a terminal:

1. The terminal is placed in an "out of service" status.

2. The Terminal Abnormal Condition program is attached to the system.

3. The Terminal Abnormal Condition program writes the error data to a destination in Transient Data Control. This destination is to be defined by the user and may be intrapartition, extrapartition, or not in existence. If the destination is not in existence, the error data is not sent as output by CICS.

4. The Terminal Abnormal Condition program then links to the user-written program (DFHTEP) to allow further terminal-oriented analysis of the error. In the Terminal Error program, the user may decide to have the terminal placed in service, have the

line placed in or out of service, have the transaction in process on the terminal abended.

The Terminal Abnormal Condition program sets defaults to these functions which the Terminal Error Program may either override or use. The facility provided to the Terminal Error program is a copy of the DECB portion of the line entry at the time of the error. The address of the line entry and a displacement to the terminal entry is provided in this copy of the line entry; the copy is then used to communicate what is to be done to the line, terminal, and task.

5. The Terminal Abnormal Condition program is detached from the system.

For information concerning error recovery on communication lines under control of TCAM, see the discussion of "TCAM I/O Recovery Procedures" in the _TCAM Programmer's Guide and Reference Manual_ (GC30-2024).

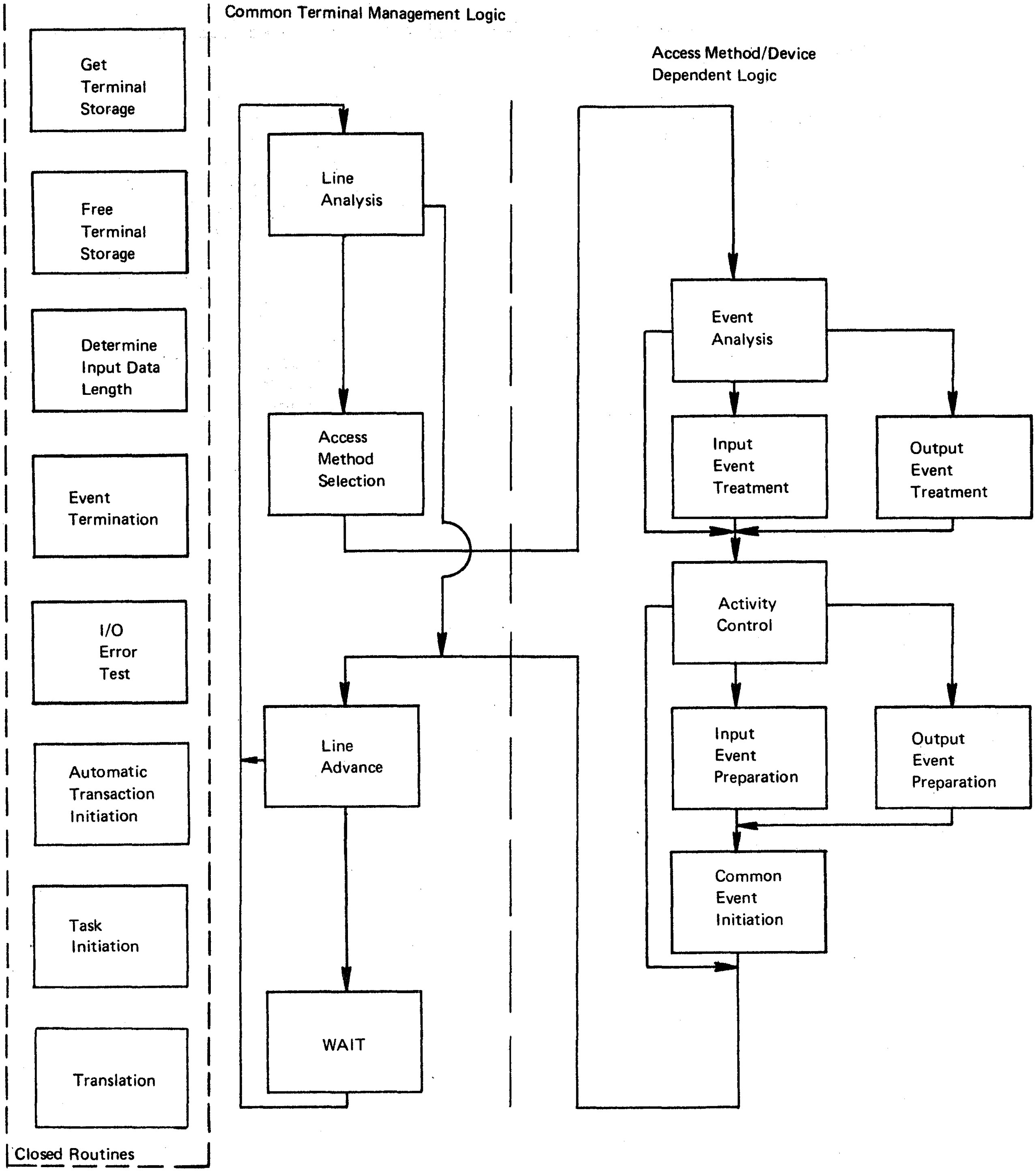


Figure 12. Terminal control program

FILE MANAGEMENT

Control of data set (data base) operations is accomplished by File Management through the File Control program. The File Control program reads from and writes to user-defined data sets, gathers statistical data, and acquires main storage for data set operations. The File Control program uses data control information defined by the user in the File Control Table generation procedure in performing input/output operations.

The File Control program uses the Indexed Sequential Access Method (ISAM) and the (Basic) Direct Access Method (DAM) to access data sets in either a random or sequential (browsing) mode. Records may be blocked or unblocked and may be fixed-length, variable-length, or undefined. The CICS/DOS systems support variable-length logical records within fixed-length blocks for ISAM data sets. Record additions to variable-length ISAM data sets are not supported, but, provided the user creates such a data set, the CICS/DOS systems will retrieve records and write updated records (with no length variation).

File Management consists of a File Control program, access method-dependent logic for each access method described to the system, and the File Control Table. User-provided data set descriptions are used to generate the File Control Table. If record segments are needed in the user-written programs, CICS provides a record segment control feature; the user must also supply record segment descriptions in order to use this feature.

Upon request, the File Control program retrieves a record and, according to segment control information referenced by the macro instructions, returns to the user only those segments of the record that are requested. The segment control feature reduces the main storage requirements for data set records during the interval of time that the transaction is being processed and may save direct access storage space if variable-length records are being used.

File Management provides the following services:

1. Randomly retrieves data from a data set.
2. Sequentially retrieves data from a data set (browsing).
3. Stores data in a data set.
4. Performs segmenting control services according to the user's specifications.
5. Indirectly retrieves data from a master data set through one or more index data sets.
6. Provides deblocking services for blocked DAM data.
7. Logically opens/closes data sets during real-time execution of CICS.
8. Provides exclusive control (within the CICS partition/region) at the logical record level for update operations on ISAM data sets, and provides exclusive control at the physical block level for DAM data sets.

A system option provides for the indirect accessing of data sets to meet the requirement of some data sets that the data be accessed by other than the major control key. A normal solution for this requirement is to establish the main data set in an indexed sequential manner to satisfy the primary indexing scheme and provide an organization that facilitates the bulk of the processing upon the data set. Additional data sets or cross indexes are created to provide access by other significant keys.

The File Control program reads the cross-index data set when directed by the user. If the cross-index entry indicates multiple entries in the main data set, information is returned to the user-written application programs to be used in selecting the appropriate main data set entry. When the cross index does not indicate multiple entries in the main data set, the File Control program reads the requested record from the main data set.

CICS optionally provides protection to the user against the concurrent update of a data set record by two or more transactions. This protection is called exclusive control. The user specifies the exclusive control option in the creation of the File Control Table. If exclusive control is specified in the File Control Table for a data set, a request for "read with update" of a logical record is queued if, and only if, another "read with update" request has been issued for that same logical record. Multiple transactions can concurrently update the same data set as long as the requests are for different logical records (ISAM) or physical blocks (DAM). Exclusive control is not provided on keyed data sets with keys larger than 225 bytes.

Another optional feature of CICS File Management is the sequential retrieval of records from the data base. This feature is known as "browsing". To initiate a browse operation, the user provides either a specific or generic (partial) record reference (key) where sequential retrieval is to begin. Each subsequent GET request by the user initiates retrieval of the next sequential record. The user also has the capability of retrieving specified segment sets while in browse mode. The application, while in browse mode, can issue random "get for update" requests without interrupting the browse operation. The application may concurrently browse several different data sets and/or multiply browse the same data set.

The File Control program acquires and disposes of the input/output areas required for its own operations through the Storage Control program.

The File Control program is used by the CICS Open/Close system service program to support the dynamic opening/closing of the user's data base data sets.

Optional user exits are provided:

1. Prior to determining what type of request for file services was issued.
2. Prior to providing the requested output service.
3. After the File Control Table is searched in response to a request for an input service.
4. Upon completion of an input event but prior to deblocking requested input records.

File Control Functions

The following functions are performed automatically by File Management. They require no communication with user-written application programs.

EVENT INITIATION: Links the appropriate access method for a read/write operation and analyzes the code that indicates the result of the operation.

EVENT SYNCHRONIZATION: Synchronizes the event in this task with all other events in the system by issuing a Task Control WAIT macro instruction. When the event is complete, control is returned to File Control to complete processing for this task.

STORAGE ACQUISITION: Acquires and initializes data set (file) storage areas for a data set operation.

DATA EVENT CONTROL INFORMATION PREPARATION: Prepares information for a data set (file) operation (the linkage to an access method).

EVENT ANALYSIS: Analyzes data set dependencies and selects the proper course of action.

FILE INTEGRITY: Provides for the automatic acquisition and disposition of exclusive control during update operations. File integrity is an optional feature.

RECORD SEGMENT SERVICES: Selects user-requested record segments and gathers all segments into a single record.

STORAGE DISPOSITION: Releases main storage file areas and exclusive record control.

The following functions are performed by File Management in response to macro instructions issued in the application program.

INITIATE BROWSING (SETL): Initiates a browse operation on the data base.

GET (READ ONLY): Retrieves a user data base record for read-only examination.

GET (UPDATE): Retrieves a user data base record for subsequent update.

GET (BROWSE): Retrieves the next sequential record in a browse operation.

PUT (UPDATE): Writes an updated record to the user's data base.

PUT (NEW RECORD): Add a new record to the user's data base.

GET WORK AREA: Allocates a File Work Area and passes it back to the application program to build a new record.

TERMINATE BROWSING (ESETL): Terminates a browse operation on the data base.

RESET BROWSING (RESETL): Allows user to skip forward or backward in the data set during a browse operation.

RELEASE EXCLUSIVE CONTROL: Provides the means of releasing exclusive control of a record which was retrieved for update but not subsequently updated.

DYNAMIC OPEN/CLOSE: Logically opens or closes specified data sets.

Data Language/I Facilities

CICS/OS provides access to DL/I data bases through either of two means: (1) a DL/I CALL formulated according to DL/I specifications, or (2) the CICS File Control instruction (DFHFC) using unique DL/I operands.

Since the CICS application program is considered a main program (as opposed to an IMS application program which is considered a subprogram), the CICS application program cannot receive PCB addresses upon entry as does an IMS application program. Therefore, at the beginning of the program, the application program must issue a special CALL to DL/I to obtain the addresses of the PCB's used in the transaction. If the DFHFC macro instruction is used to do this, the transaction can dynamically specify the name of the PSB to handle the call. This allows several users with their individual data bases to share a transaction; for example, each of several hospitals can access its own data base through the use of a shared transaction.

To keep the application program quasi-reentrant, main storage must be dynamically acquired for the DL/I work area and for Segment Search Arguments (SSA's), if they are to be used. If a DL/I CALL is issued, the application program must acquire DL/I work area storage for GET-type (Gxxx) and ISRT calls. If a DFHFC macro instruction is issued, storage need only be acquired for ISRT calls; CICS acquires the storage required for Gxxx calls. In all cases, the application program must acquire storage for SSA's if they are to be used in a CALL and must build the SSA's in this storage.

The DL/I CALL and/or DFHFC macro instruction are processed by CICS-DL/I interface routines which acquire work area storage (if necessary), pass the parameter list to DL/I routines, and return the requested segment (data) and/or return codes to the application program. If IMS would have terminated the transaction with a pseudo-abend, the interface routines will terminate the transaction with a CICS ABEND and the IMS ABEND code. Other transactions are not affected.

Formulation of DL/I calls is described in the IMS Application Programming Reference Manual (SH20-0912). Use of the DFHFC macro instruction for DL/I access is described in the CICS Application Programmer's Reference Manual (SH20-1047). For further information on DL/I facilities, refer to the appropriate IMS/360 publications as listed in the "Bibliography".

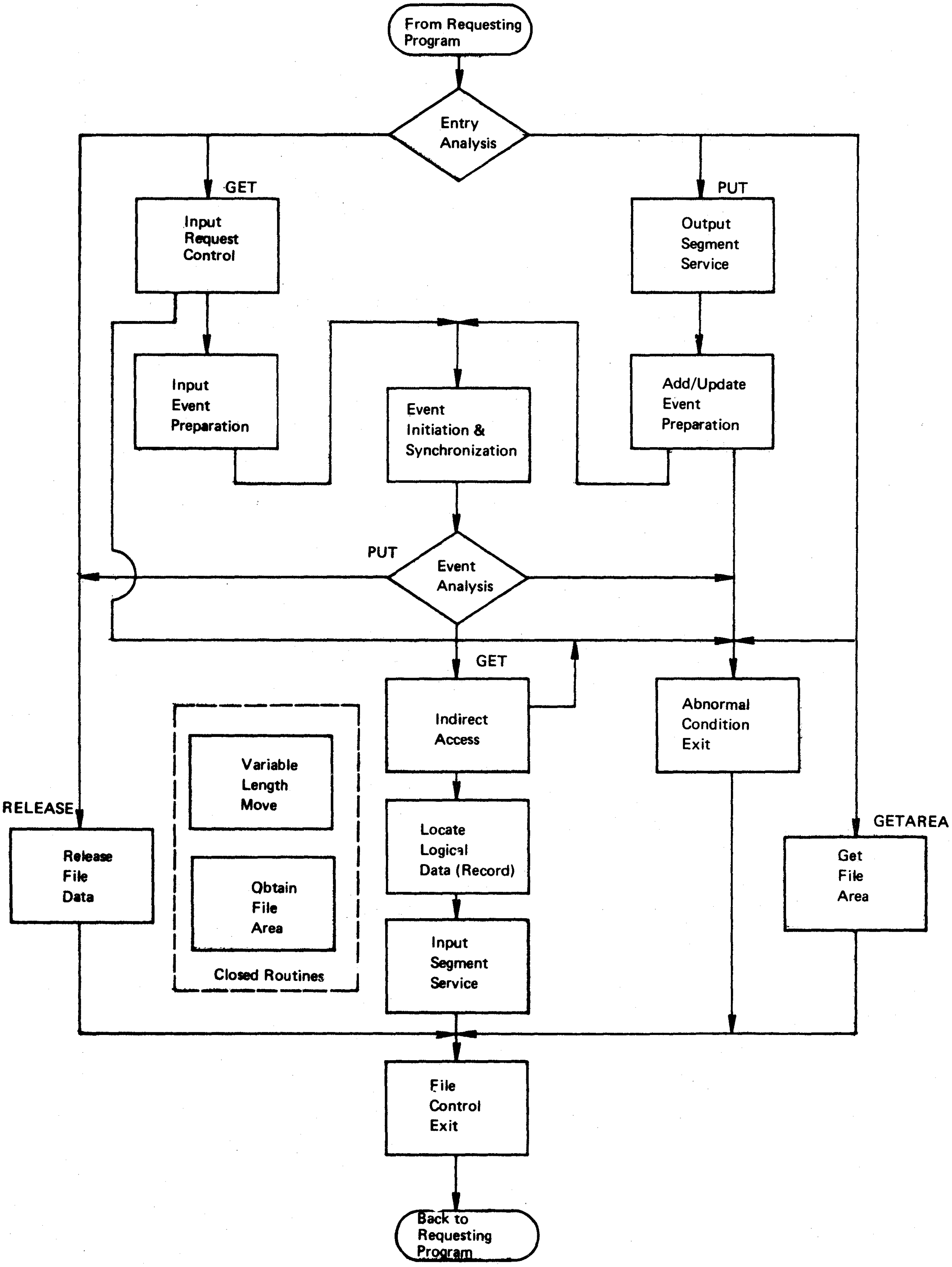


Figure 13. File control program

TRANSIENT DATA MANAGEMENT

Transient Data Management provides a generalized queuing facility where data can be queued (stored) for subsequent internal or external (offline) processing. Selected units of information, as specified by the user through use of CICS macro instructions, can be routed to or from predefined symbolic destinations, either intrapartition or extrapartition.

Intrapartition destinations are queues of data on direct access devices developed for input to one or more CICS transactions. Intrapartition destinations are accessible only by CICS transactions within the CICS partition/region. Data directed to or from these internal destinations is called intrapartition data and may consist only of variable-length records. Intrapartition destinations can be associated with (1) a terminal, (2) an output data set, or (3) an application program under the control of CICS.

The intrapartition queue is reusable; an option permits the user to indicate, by symbolic destination, if Transient Data Space Management is to control the reuse of tracks in a particular DESTID or whether the user will control the releasing of space through the transient data PURGE macro facility. It should be remembered that if Transient Data Space Management is not used, these queues continue to grow, irrespective of whether the data has been read, until the user purges them.

Examples of the data queued for intrapartition processing are:

1. Data or administrative messages to be switched to output terminals.
2. Transaction-related data for which the output terminal is different from the input terminal.
3. Transactions which require processes that are to be performed serially, not concurrently. An example of this type of process is one in which pending order numbers are to be assigned.
4. Data to be used in a file update which could pass through the queue to allow the data to be applied in sequence.
5. Batched input data to be processed asynchronously.

Extrapartition destinations are data sets that are external to the CICS partition/region, residing on tape or direct access devices. Data directed to or from these external destinations is called extrapartition data and may consist of sequential records that are fixed or variable-length, blocked or unblocked.

Examples of data placed on extrapartition data sets are:

1. System statistics.
2. Transaction error messages.
3. A complete log of all transactions.
4. Customer data such as cash payments which can be applied offline.

Intrapartition and extrapartition destinations can be referenced through indirect destinations. This facility provides some flexibility in program maintenance in that an installation can be changed, giving a destination a new symbolic name, without recompiling existing programs. These programs can be allowed to route data to the previously existing symbolic name; however, the previously existing symbolic name is now an indirect destination that refers to the new symbolic name.

When data is sent to an intrapartition destination and the number of entries (PUT's from one or more programs) in the queue reaches a predefined level (trigger level), the transaction is automatically initiated to process the data in that queue. If no task has been created to retrieve the data from the queue, Transient Data Control establishes the conditions for automatic transaction initiation.

The automatic transaction initiation facility allows a user transaction to be initiated either immediately, if the ultimate data destination is a data set, or, if the ultimate data destination is a terminal, when that terminal has no task associated with it. The terminal processing status must be such that messages may be sent to it automatically. The destination and the transaction identifications are specified in the Destination Control Table.

Through the trigger level and automatic transaction initiation facility, a user-written application program has the capability to switch messages to terminals. Once a task has been initiated, a macro instruction in the user-written application program is executed to retrieve the queued data. All data in the queue is retrieved sequentially for the user's application program. The user's program can be written to make as many requests as necessary to retrieve all the data in a destination queue.

Optional user exits are provided:

1. After locating the appropriate entry in the Destination Control Table but prior to writing data in response to an output request.
2. After acquiring data in response to an input request.
3. Prior to determining what type of request for transient data services was issued.

Transient Data Control Functions

The following functions are performed by Transient Data Management in response to CICS macro instructions issued in user-written application programs.

INTRAPARTITION DATA DISPOSITION: Controls and queues data for serially reusable or reentrant facilities (programs, terminals) related to this partition/region.

INTRAPARTITION DATA ACQUISITION: Retrieves data that has been placed in a queue for subsequent internal processing.

EXTRAPARTITION DATA ACQUISITION: Enters a sequentially organized data set into the system.

EXTRAPARTITION DATA DISPOSITION: Writes fixed- or variable-length data in a blocked or unblocked format on sequential devices for offline processing.

AUTOMATIC TRANSACTION INITIATION: Initiates a transaction to process previously queued transient data when a predefined trigger level is reached.

DYNAMIC OPEN/CLOSE: Logically opens or closes specified extrapartition data sets (destinations) during the real-time execution of CICS.

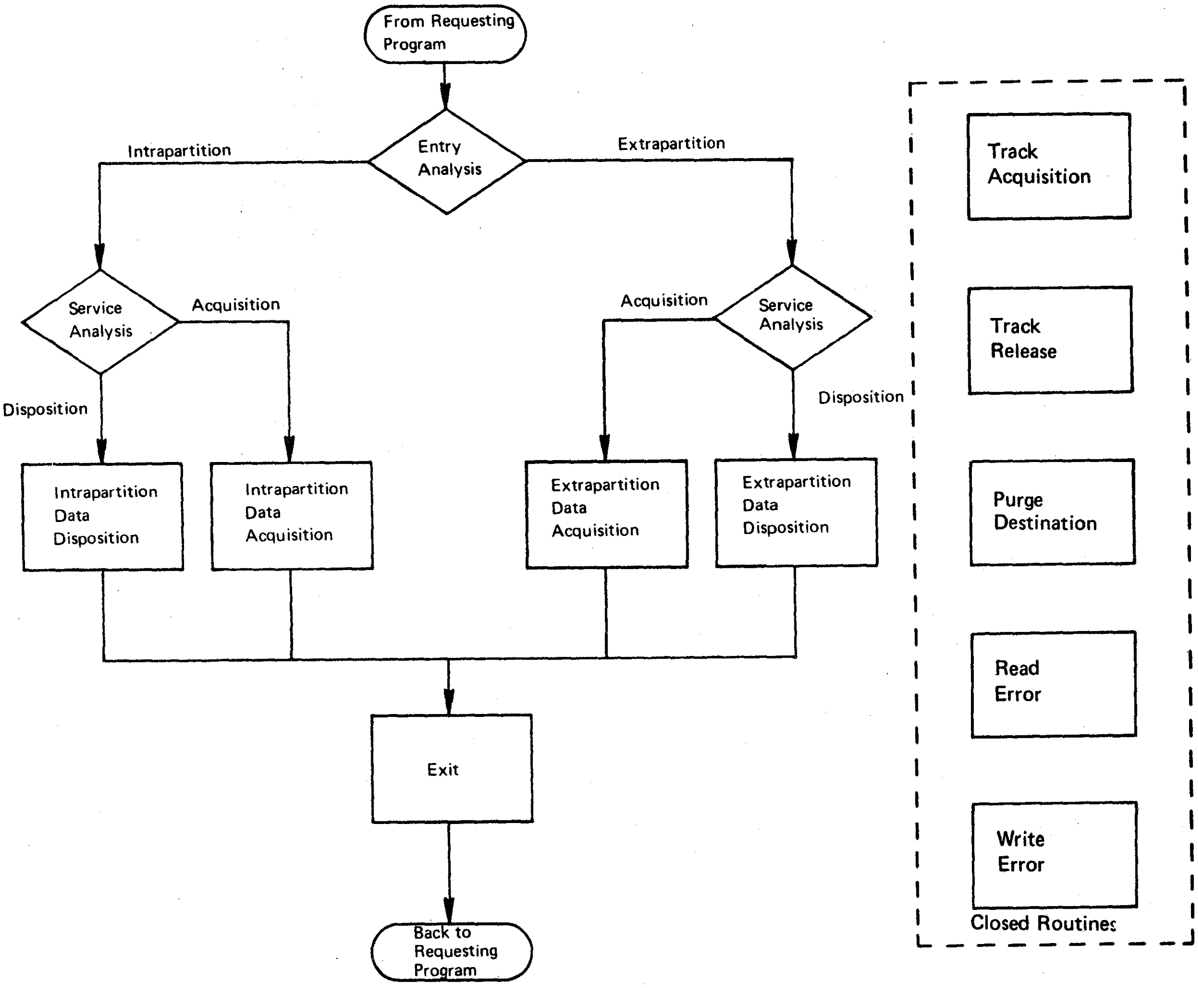


Figure 14. Transient data control program

TEMPORARY STORAGE MANAGEMENT

The processing needed to save data temporarily in main storage or on a direct access storage device (DASD) and to retrieve it at a later time is performed by Temporary Storage Management. It may be desirable to store data temporarily during the processing of a transaction to conserve main storage. Or when a transaction must be suspended, certain data may have to be saved for later processing by the resumed transaction.

The Temporary Storage Control program provides a "scratch pad" to be used by user-written application programs (transactions). The user might use the temporary direct access storage to accumulate data during a transaction which has multiple inputs from the terminal. Although the above data could be accumulated in main storage, the user can conserve main storage by placing the transaction data on the direct access storage device until the end of the transaction. In such cases, the Temporary Storage Control program (1) puts the data on direct access storage using BDAM to write it, or (2) if requested, saves the data in main storage. The Temporary Storage Control program retrieves the data on request.

If the release of data is requested, the Temporary Storage Control program frees the main storage or direct access storage space that was used for the data.

The Temporary Storage Control program, through its management of direct access storage space, allows the user to provide a "paging" capability often used with graphic terminals. The user may create several pages of data in response to an inquiry from a graphic terminal, PUT these pages to temporary storage destinations which have unique identifications, and then retrieve these pages as the terminal operator requests forward or backward paging.

Temporary Storage Management Functions

The following functions are performed by Temporary Storage Management in response to CICS macro instructions issued in user-written application programs.

TEMPORARY STORAGE ACQUISITION: Provides main storage or direct access space and saves specified data in this space.

TEMPORARY STORAGE DISPOSITION: Locates data being saved in a temporary storage area, puts it into a user-specified main storage location, and releases the space which the data occupied.

TEMPORARY STORAGE RELEASE: Locates data being saved in a temporary storage area, removes it from the Temporary Storage Control chain, and releases the space occupied by the data.

Optional user exits are provided:

1. Prior to writing data in response to an output request.
2. Prior to returning control to the application program after servicing an input request.
3. Prior to determining what type of request for temporary storage services was issued.

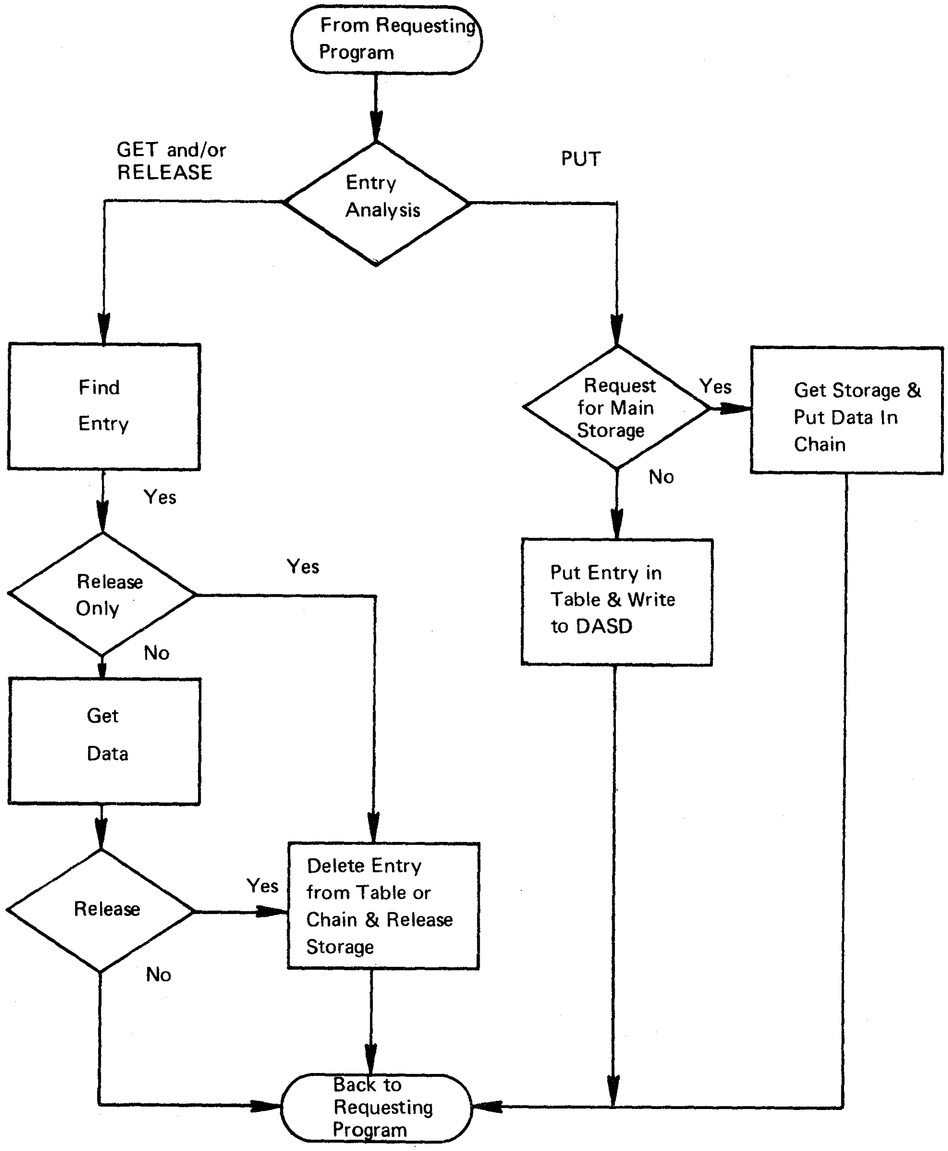

From Requesting Program
GET and/or RELEASE
Entry Analysis
PUT
Find Entry
Yes
Request for Main Storage
Yes
Get Storage & Put Data In Chain
No
Release Only
Yes
Put Entry in Table & Write to DASD
No
Get Data
Release
Yes
Delete Entry from Table or Chain & Release Storage
No
Back to Requesting Program

Figure 15. Temporary storage program

REAL-TIME ENVIRONMENT DEFINITION

The operating environment required for real-time processing is defined and controlled by user-generated system control tables and system service tables. Because CICS allows the user to maintain a number of versions of tables (and programs) through the use of suffix characters, the real-time environment is highly flexible and easily managed.

SYSTEM CONTROL TABLES

The system control tables are the means whereby the user identifies each element to be included in the real-time environment, the attributes of each element, and the treatment he wishes given to each element of that environment. Several of the tables are considered essential while the others are optional and, like the CICS management programs, are dynamically configured into the system under user control during system initialization.

The System Control Tables considered essential to the operation of CICS are:

1. Program Control Table (PCT)
2. Processing Program Table (PPT)
3. Terminal Control Table (TCT)

The System Control Tables considered optional to the operation of CICS are:

1. File Control Table (FCT)
2. Destination Control Table (DCT)

Program Control Table (PCT)

The Program Control Table is used to define the transaction environment. An entry must be included to identify each transaction code which is to be validly invoked by any source within the system. Additional attributes applicable to each transaction are also included in the table entry. For example:

1. Transaction priority and security identification (three-position decimal values)
2. Transaction Work Area (TWA) length used to determine the size of the TWA to be acquired for this transaction.
3. Initial processing program identification
4. Transaction statistics
5. Transaction purge status (purgeable or nonpurgeable)
6. Transaction type (synchronous or asynchronous)

Processing Program Table (PPT)

The Processing Program Table is used to define the application (processing) program environment. There must be an entry in the table for each application program, thereby signifying that the application program is valid for execution within CICS and that the program is located in the CICS Real-Time Program Library. Also included in the Processing Program Table are attributes of the application program such as:

1. Program source language (Assembler language, ANS COBOL, or PL/I)
2. Program residency status (resident or nonresident)

Terminal Control Table (TCT)

The Terminal Control Table is used to define the user's terminal environment. There must be an entry in the table to describe each:

1. Communication line group/control unit
2. Communication line
3. Terminal
4. TCAM process queue

Parameters applicable to each communication line group/control unit include:

1. Device address
2. Error recovery/recording options
3. Special device-dependent features (optional)
4. Binary synchronous device options
5. Class (conversational or batch)

Parameters applicable to each communication line include:

1. Access method
2. Terminal types associated with the line
3. Length of input message area
4. Various device-dependent options and features

Parameters applicable to each terminal include:

1. Symbolic identification of each terminal
2. Terminal priority
3. Terminal type
4. Terminal address
5. Initial terminal status

File Control Table (FCT)

The File Control Table is an optional table used to describe to the CICS File Control program the user's data base (data set) environment. There must be one entry in the table for each data set the user wishes to access during execution of CICS. In addition, to provide a symbolic identification for each data set the following attributes may also be specified:

1. Data set organization (ISAM, DAM, or DL/I)
2. Accessing options
3. Data set characteristics (for example, block size)
4. Indirect accessing or indexing control information
5. Record segment definitions

Destination Control Table (DCT)

The optional Destination Control Table (DCT) is used to describe to CICS the characteristics of data to be processed by the Transient Data Control program. There must be an entry in the table for each extrapartition data set and for each extrapartition or intrapartition symbolic destination.

If system statistics and error conditions are to be logged by CICS, the appropriate destinations must be referenced by entries in the DCT. If the DCT is omitted, system statistics and error conditions are not logged.

Extrapartition data is generated either externally to the CICS environment and processed within CICS, or it is generated within CICS and processed externally (for example, batched mode).

Intrapartition data is generated and processed within the CICS environment.

In addition to providing a symbolic identification for each destination, the following information is included in each entry to the Destination Control Table:

1. Destination characteristics (for example, block size)
2. Automatic transaction initiation information
3. Indirect destination specifications

SYSTEM SERVICE TABLES

The system service tables (optional) provide the user with increased control over the operation of CICS. The system service tables are:

1. Sign-on Table that contains the operator security and priority data to permit an operator to sign on the system.
2. Terminal List Tables that contain, by groups, the identification of terminals that are related logically; for example, the terminals in a remote location or all terminals whose operators are under the control of a supervisor. A supervisor, through his table, may alter the status of a terminal under his control. For example, he can place a terminal in an out-of-service condition.

The Sign-on Table and the Terminal List Tables reside on a direct access storage device and are called into main storage as required.

Sign-on Table (SNT)

The Sign-on Table provides the means for permanently retaining terminal operator data. It is accessed when a terminal operator initiates the sign-on procedure.

During the sign-on procedure, the name of the terminal operator is entered at the terminal and is used to locate the appropriate operator entry in the table. The operator entry in the table contains data used to verify the operator name and to establish a priority and a security key for the transactions which the operator subsequently enters.

The priority value related to the operator is used to develop the task priority for processing a transaction. The operator's security key is used in a security check of all transactions subsequently entered. The security key which is placed in the appropriate Terminal Control Table entry for the operator is matched with the transaction security contained in the transaction's Program Control Table (PCT) entry.

If the operator security key contains any position that matches the security value in the PCT entry, the transaction is accepted. Otherwise, a security check has occurred and the transaction is terminated. A security key of 1 is the default option in the creation of the Sign-on Table and the Program Control Table and is present in the Terminal Control Table until altered by a sign-on procedure. The security key default option allows transactions with a transaction security of 1 to be entered into the system by the operator without the sign-on procedure.

Terminal List Tables (TLT)

The Terminal List Tables provide the user with the means of retaining the terminal identifications in logical groupings. A logical grouping of terminals could be all the terminals that are under the control of a supervisor. Each supervisor might have a terminal list table. All terminals which serve a similar function represent a logical grouping. The user might create a grouping of terminal identifications to facilitate the dispatching of general messages to terminals.

The system service programs of CICS use Terminal List Tables to perform terminal status change operations. A unique Terminal List Table must be created for each supervisor who is to have the ability to alter the terminal status of any terminal under his control. Any operation which is to change the status of an entire group of terminals requires a Terminal List Table containing the identifications of all terminals in that group. A unique identification is assigned to each table by the user.

REAL-TIME SYSTEM SERVICE FUNCTIONS

Included as part of CICS are a number of ancillary application level processing programs identified as the system service functions. Though several are considered optional to the system, the functions and services of these programs are believed vital to the successful implementation of a data base/data communication system. The system service functions include the following:

1. Sign On/Sign Off
2. Master Terminal
3. Supervisor Terminal
4. Operator Terminal
5. System Statistics
6. Abnormal Condition
7. Terminal Abnormal Condition
8. System Termination
9. Asynchronous Transaction Input Processor
10. Asynchronous Transaction Output Processor
11. Trace
12. Dynamic Open/Close
13. Time of Day Control

SIGN ON/SIGN OFF

The sign-on/sign-off function is optional to the user. It can be used to varying degrees if selected as an option; for example, only master terminal operators may be required to sign on/sign off. The function provides system security to the degree it is employed.

The sign-on security check is made on the name and password parameters supplied by the terminal operator; both must be present, correctly entered, and must agree with an existing entry in the Sign-on Table. If the verification is positive, the operator conditions are set to indicate the operator is signed on.

The sign-off request causes the operator conditions to be set to the signed-off status and statistics are extracted. In signing off, the operator can optionally request the removal of the terminal from the polling list.

MASTER TERMINAL

Individuals within the organization may be defined in the Sign-on Table as having authority to perform master terminal functions. The operator may sign on with any terminal, causing that terminal to be designated as the master terminal. It is from the master terminal that the overall CICS operation is controlled. The master terminal operators can dynamically vary system parameters as well as change the status of lines and/or terminals and/or data sets. All processing for master terminal operation is controlled by conversational terminal interaction.

SUPERVISOR TERMINAL

Individuals within the organization may be defined in the Sign-on Table as having authorization to perform Supervisory functions. He may sign on at any terminal, causing that terminal to be designated as a supervisory terminal. The supervisor can change the status and/or the processing mode of his terminal or any terminal under his supervision.

OPERATOR TERMINAL

Except for the master terminal operator or supervisor, a terminal operator is able to control the status and processing mode of only his terminal.

SYSTEM STATISTICS

System statistics are maintained by the various CICS management programs during the execution of CICS. These statistics can be displayed during the day in part or in their entirety on the request of any terminal operator whose security code allows the request of such information.

System statistics are transmitted to the Transient Data destination CSSL as variable-length, unblocked records (maximum block size equal to 136) when the system is normally terminated or when requested by the terminal operator.

For further details, see the section "Control, Audit, and Reconstruction Procedures".

ABNORMAL CONDITION

The Abnormal Condition program is used by Program Control to analyze the type of abnormal condition that exists. Appropriate action is taken. For example, the terminal operator is notified that a task has been terminated, and appropriate messages are logged for destination CSMT. Statistics are gathered for the number of transaction errors.

TERMINAL ABNORMAL CONDITION

The Terminal Abnormal Condition program is used by Terminal Control to analyze the type of abnormal condition that exits. Appropriate action is taken with regard to terminal statistics, line statistics, terminal status, and line status; the task (transaction) may be terminated. Messages are logged to the Transient Data master terminal destination (CSMT) or terminal log destination (CSTL). A link is provided to the user-written Terminal Error program to allow the user to attempt recovery from transmission errors and to allow the task to continue processing.

SYSTEM TERMINATION

System termination provides for the orderly shutdown of CICS operation. Terminals are allowed to quiesce, operators are notified that termination is in process, data sets are closed, and system statistics are extracted.

ASYNCHRONOUS TRANSACTION PROCESSING

The optional Asynchronous Transaction Processing facility (ATP) allows transactions and the data associated with those transactions to be batched for asynchronous (concurrent) processing. The ATP facility also allows control of the asynchronous batch processing and of the writing of data generated by that processing.

Application programs may be processed either synchronously (associated with a terminal) or asynchronously (batched via ATP) depending upon how they are invoked by the terminal operator. Thus application programs need not be specially written (or rewritten) to use the ATP facility.

After the entry of a batch of transactions is completed by the terminal operator, that terminal is freed for other work while the transactions are being processed. Output generated by the processing of batched transactions is queued (stored) for subsequent transmission to a specified terminal; transmission occurs upon request of the terminal operator. The output can be directed to the originating terminal cr to another terminal.

The ATP facility is primarily designed for, but not limited to, high-speed data entry terminals such as the 2780 Data Transmission Terminal.

TRACE

The optional trace function provides a main storage Trace Table containing entries that reflect the execution of various CICS macro instructions by user-written application programs and CICS management programs. The trace facility is especially useful in a program testing and debugging environment. The trace function, if generated, can be turned on or off through the CICS Master Terminal facility.

DYNAMIC OPEN/CLOSE

This optional facility allows the user to dynamically open/close his data sets as often as desired during the real-time execution of CICS. For example, this makes it possible for the user to defer the cpening of data sets at System Initialization and open/close them later as they are needed. The dynamic open/close capability is applicable to File Management (data base data sets), Dump Management (dump data sets), and Transient Data Management (extrapartition data sets).

TIME OF DAY CONTROL

This optional facility makes it possible for the user to operate CICS on a continuous round-the-clock basis. When the time of day maintained by the operating system is changed either by the operating system (for example, OS resetting the clock to zero at midnight) or by the console operator, CICS recognizes the situation where a "negative" change in the time of day has occurred and (1) adjusts expiration times maintained by CICS to reflect the negative value and (2) resets its time of day to the time of day maintained by the operating system.

FUNCTIONS NOT REAL-TIME

Functions that are not real-time are used to support the system in real-time execution. System Initialization is not a real-time function but is required to prepare for and bring the system into real-time execution. The Dump Utility function is not a real-time function but is used to format and print the main storage images collected.

SYSTEM INITIALIZATION

The System Initialization program is used to bring CICS into real-time execution.

The System Initialization program is resident only long enough to perform the startup sequence which includes the following major steps:

1. Establish initialization parameters - Provide the user with the capability of dynamically configuring a CICS nucleus through use of the System Initialization Table and to specify various system control parameters which affect the system performance.
2. Load selected CICS management programs and tables to form the CICS nucleus.
3. Open CICS system data sets and user data sets.
4. Establish the main storage boundaries for the CICS Storage Control program.
5. Transfer control to an entry point in the CICS Nucleus which in turn branches to Terminal Control to begin polling.

The main storage occupied by System Initialization becomes part of the dynamic storage area.

HIGH-LEVEL LANGUAGE PREPROCESSOR

The High-Level Language Preprocessor program prepares a high-level language program for input to the Assembler. The Assembler then generates the high-level language statements for CICS macros for input to the High-Level Language Compiler.

CICS/DOS MAINTENANCE AND LINKAGE EDITOR

The CICS/DOS Maintenance and Linkage Editor program maintains the CICS Real-Time Relocatable Program Library. This program combines the functions of the DOS Librarian and Maintenance and Linkage Editor programs for support of the CICS Real-Time Relocatable Library. It can be executed by itself to perform maintenance functions or as a step following a language translator to perform maintenance functions and/or link edit.

The functions supported are:

1. Create
2. Copy
3. Delete
4. Condense
5. Link Edit
6. Display

DUMP UTILITY

The output from the Dump Control program, extracted during the execution of CICS, is formatted and printed by the Dump Utility program. This program operates in batch mode while one of the dump data sets is closed. Each area, program, and table entry is identified, formatted, and printed separately with both actual and relative addresses to facilitate faster analysis.

CUSTOMER INSTALLATION REQUIREMENTS

The specific installation requirements must be determined by the customer based on his needs and available resources. He must meet the requirements as presented in the "Programming Systems" section and "System Configuration" section of this publication.

CUSTOMER RESPONSIBILITIES

Before completing the installation of CICS, the customer must:

1. Order and satisfactorily install all required communication equipment for the initial system.
2. Have a thorough knowledge of the information system application.
3. Install an operating system (for example, OS/360 or DOS/360).
4. Train system analysts, programmers, and operators in the use of CICS.
5. Design and create master data sets.
6. Design terminal formats.
7. Develop a terminal-oriented application training program for terminal operators.
8. Develop, write, and test the inquiry and transaction processing programs using CICS macro facilities.
9. Develop procedures to ensure adequate security for data in the system.
10. Develop backup and recovery procedures for the information system application.
11. Develop terminal conversion procedures and schedules.
12. Select desired system options beyond those necessary for the operation of CICS.
13. Write a Message Control program (MCP) if TCAM is to be used.
14. Study the CICS Terminal Error program (DFHTEP) during the system design phase to determine whether additional or alternate terminal error handling is required (the user must provide error handling routines to handle terminal errors other than I/O errors, or to handle I/O errors in a manner different from the way they are handled by DFHTEP).

Successful implementation of an online/real-time system is a complex undertaking which is made easier through use of CICS. However, to help ensure a smooth and successful installation of CICS, it is recommended that:

1. At least one person in the installation have a good knowledge of Assembler language.
2. The system programmer(s) have a thorough understanding of the characteristics of the terminals to be used (including some BTAM knowledge and/or experience).
3. A thorough and effective implementation plan be developed prior to installation of CICS which details the necessary installation steps and provides time estimates for each step. See your IBM representative and/or systems engineer for assistance in developing such an implementation plan.

PERFORMANCE SPECIFICATIONS

CICS provides a functional capability. Since it is the user's responsibility to prepare the system configuration, provide the operational environment, and design and implement the application program(s) to meet his requirements, it is difficult to provide other than a range of possible performance objectives. Performance within the range is predicated on the user having reasonably standard requirements and the user implementing his application with adherence to good real-time programming practices.

TIMING AND THROUGHPUT INFORMATION

As the users' demands on the system increase, that is, through transactions per unit time or through added function, the response time may be held relatively constant by increasing the amount of main storage, channel throughput, and CPU power. Because of the techniques employed in the CICS/DOS-ENTRY system to operate in a partition of minimum size, the operational reaction to increasing these resources is not as dramatic as in the CICS/DOS-STANDARD system or the CICS/OS-STANDARD system.

Response time varies depending upon:

1. The type of terminal. The IBM 2260 Display Station has a substantially faster transmission rate in characters per second than a terminal that produces hard copy such as the IBM 2740 Communication Terminal.
2. The number of terminals on the system. Each terminal represents an increase in use of system resources.
3. The number of terminals on a communication line and the transmission speed of the line. Multiple terminals on a single communication line increase the chance of a delay in communication with a specific terminal; transmission speed of the line directly affects response time.
4. The terminal operator entry technique and the output response format. Inquiry and order entry dictate the number of transmissions to and from a terminal and the total number of characters that are transmitted each time.
5. The central processing unit and related input/output devices. The resources available on the data processing system as well as those available at any one moment to process multiple transactions affect the throughput rate for each transaction.
6. The scope of the user application. The demands upon the entire system increase as the application is extended from one in which inquiries are processed to one in which: (1) orders are entered, (2) data sets (files) are updated to reflect change, and (3) work assignments are distributed.
7. The number of transactions to be entered. As the volume of transactions increases, the likelihood that there will be a wait for the availability of a facility such as a data set also increases.

Performance can be determined only by detailed system design and analysis.

Note: When binary synchronous communication lines are part of the user's configuration, it is possible for these communication lines to time out if "excessive" CPU time is required by the application program. The user can alleviate this condition in either of two ways: (1) by having the application program issue a CICS WAIT macro instruction to voluntarily relinquish control, or (2) by setting the runaway task time interval to a value that causes the long-running task to be terminated before the line time out can occur.

An additional consideration would be use of the Asynchronous Transaction Processing facility to collect transactions and data with a minimum of overhead and to have the transactions processed asynchronously with other terminal activity.

CONTROL, AUDIT, AND RECONSTRUCTION PROCEDURES

The creation of an audit trail is the user's responsibility. Through the use of Transient Data Control, the user can output and log transaction data to a sequential file. The user describes symbolically for CICS the audit data set and issues from his program to the Transient Data Control program a CICS macro instruction that causes transaction data to be stored in that data set when each transaction is completed. The same data set can be organized to contain not only the transaction log but other control and summary data. The user is responsible for the review and control of the data on the log.

If DL/I data bases are used in conjunction with CICS/OS, the DL/I recovery utilities (which are included in the IMS Data Base system) are available for data base back up, reconstruction, and repair. During execution of CICS/OS with DL/I, the DL/I change logging function produces a data set which is an audit trail of changes made to DL/I data bases. In addition, the CICS-DL/I Interface routine (which acts as an IMS application program) logs a termination record when each transaction that alters a DL/I data base is terminated. Since it contains these termination records, the log data set resembles the log data set produced by the IMS online system and can be used with the DL/I Data Base Utility to back out the changes made by any transaction not terminated. The Data Base Recovery Utility can be used to recover a data base that has become unusable due to hardware failure.

CICS maintains statistical data about system events. As a part of closedown procedures, the statistical data is written to a user-specified sequential output device. Statistical information maintained by CICS includes:

1. Number of tasks in system for any time period
2. Number of tasks initiated
3. Number of storage acquisitions
4. Number of times storage cushion is released
5. Number of times storage request is queued
6. Number of times storage queue is established
7. Maximum number of requests in storage queue
8. Number of times a program is used
9. Number of READ requests per data set (file)
10. Number of WRITE update requests per data set
11. Number of WRITE add requests per data set
12. Number of READS from overflow area per data set (ISAM data sets only)
13. Number of WRITES (per data set) to extrapartition data set
14. Number of WRITES (per data set) to intrapartition data set
15. Number of input messages per terminal
16. Number of output messages per terminal
17. Number of transmission errors per terminal
18. Number of transactions
19. Number of transaction errors
20. Number of polls issued per line
21. Number of input messages per line pool
22. Number of output messages per line pool
23. Number of transmission errors per line pool
24. Number of transactions per line pool
25. Number of transaction errors per line pool

Note: Items 21-25 apply only to lines controlled by TCAM under CICS/OS, where POOL=YES has been specified in the DFHTC TYPE=LINE macro instruction. Item 20 is not applicable to lines controlled by TCAM.

Statistical information is necessary if management is to plan effectively for future system growth and if the actual system interactions are to be understood. The statistics help the system programmer determine that proper data set allocation has been made, identify programs with little demand which should not remain core resident, determine the activity from each terminal, and, in general, determine that the resources of the system are being effectively used.

All CICS management programs are coded using System/360 Assembler language. Communication with CICS occurs via the CICS macro instructions and the coding which is included in the user-written programs. All normal diagnostic and serviceability aids as provided by the operating systems are utilized as applicable.

CICS operates as a single task within a partition/region and may operate in a dedicated or multiprogramming environment. The selection of the environment is the user's responsibility, as is the selection of system options beyond those required for the operation of CICS.

If access to the Data Language/I (DL/I) facility of the IMS/360 Version 2 Data Base System is provided by the CICS/OS-STANDARD system, the CICS-DL/I interface operates as a separate task within the CICS partition/region. In this case, the user must have a multiprogramming environment.

The CICS/DOS-ENTRY and CICS/DOS-STANDARD systems operate under the IBM System/360 Disk Operating System (DOS/360). The following components of DOS/360 are required:

- Resident Supervisor: (2311) 360N-SV-474 or (2314) 360N-SV-486
- System Control and Basic IOCS, 360N-CL-453
- Direct Access Method, 360N-IO-454
- Basic Telecommunications Access Method, 360N-CQ-469
- Assembler, 360N-AS-465 (14K variant) or Assembler F, 360N-AS-466
- Utilities Group 1, 360N-UT-461
- Utilities Group 2, 360N-UT-462
- Consecutive Tape IOCS, 360N-IO-456 and/or Consecutive Disk IOCS, 360N-IO-455

The Interval Timer Option, Decimal Arithmetic Feature, and BTAM Teleprocessing feature must be included in the DOS system generation.

In addition to the above DOS/360 components, the user may require any of the following:

- Indexed Sequential File Management System, 360N-IO-457
- Full ANS COBOL V3 Compiler, 5736-CB2, and Full ANS COBOL Library, 5736-LM2 (5736-LM2 does not include support for object code produced by the optimizing option)
- ANS COBOL Subset (DOS), 5736-CB1
- ANS COBOL, 360N-CB-482
- PL/I Optimizing Compiler (DOS), 5736-PL1
- PL/I Resident Library (DOS), 5736-LM4
- PL/I Transient Library (DOS), 5736-LM5

Note: The user should be aware that the ANS COBOL Compiler (360N-CB-482), Full ANS COBOL V3 Compiler (5736-CB2), PL/I Optimizing Compiler (5736-PL1), and Assembler F (360N-AS-466) require a main storage partition larger than the minimum required for operation of the CICS/DOS-ENTRY system. He should also be aware that if the ANS COBOL Compiler (360N-CB-482) or the Full ANS COBOL V3 Compiler (5736-CB2) is used with the minimum main storage size for the CICS/DOS-STANDARD system, there is space for only a 10K DOS nucleus.

CICS/OS-STANDARD V2 operates under the IBM System/360 Operating System (OS/360). The following components of OS/360 are required:

- Supervisor: MFT, 360S-CI-505, or MVT, 360S-CI-535
- Primary Data Management, 360S-DM-508
- Direct Access Method (BDAM), 360S-DM-509
- Basic Telecommunications Access Method (BTAM), 360S-CO-513, and/or Graphic Programming Services, 360S-IO-523, and/or Telecommunications Access Method (TCAM) Level 4, 360S-CQ-548
- Assembler F, 360S-AS-037, and/or Assembler H, 5734-AS1
- Linkage Editor (E), 360S-ED-510, or Linkage Editor (F), 360S-ED-521
- Utilities, 360S-UT-506

The Multiple WAIT and Interval Timer options must be included in the OS system generation.

In addition to the above OS/360 components, the user may require any of the following:

- Indexed Sequential Access Method (ISAM), 360S-IO-526
- Full ANS COBOL V3 Compiler and Library, 5734-CB1
- Full ANS COBOL V4 Compiler and Library, 5734-CB2
- ANS COBOL, 360S-CB-545, and ANS COBOL Library, 360S-LM-546
- PL/I Optimizing Compiler and Libraries, 5734-PL3
- PL/I F, 360S-NL-511, and PL/I F Subroutine Library, 360S-LM-512
- 3735 Form Description Macros and Utility, 360S-CQ-596
- A Type 4 SVC Number to be assigned to CICS for support of the 7770 Audio Response Unit
- IMS (Version 2, Modification Level 2 or later) Data Base System (5734-XX6) and OS system generation options required to handle an IMS Data Communication System.

Note: To use the optional "browsing" feature of CICS File Management, the CICS/OS user must have an operating system at least as current as Release 20.1 of OS/360; the CICS/DOS user with ISAM must use the CICS ISAM logic module.

To use the optional dynamic open/close function, the CICS/OS user must have an operating system at least as current as Release 20.0 of OS/360.

SYSTEM CONFIGURATION

The minimum processing unit for the CICS/DOS-ENTRY system is a 2025 Model ED (48K).

The minimum processing unit for the CICS/DOS-STANDARD system is a 2030 Model F (64K).

The minimum processing unit for the CICS/OS-STANDARD V2 system is a 2040 Model G (128K) using OS/360 MFT, or, a 2040 Model H (256K) using OS/360 MFT or MVT.

Unless incorporated as standard features on the processing units, the Decimal Arithmetic (#3237) and Interval Timer (#4760) features are required. The configuration must include sufficient I/O devices to support the requirements for: system output, system residence, and system data sets. Sufficient direct access storage must be provided to satisfy user information storage requirements and may consist of 2311 Disk Storage Drives and/or 2314/2319 Direct Access Storage Facilities and/or 2321 Data Cell Drives and/or 3330 Disk Storage.

The appropriate line adapters and telecommunications control units must be included in the system configuration.

Distribution and maintenance of CICS requires the availability of either one 9-track or one 7-track (with Data Conversion feature) tape drive. CICS/DOS is distributed on 9-track or 7-track tape, or, cptionally, on a 2316 Disk Pack.

ANS COBOL programs will execute under CICS on the minimum CICS/DOS-ENTRY configuration. However, the ANS COBOL compiler, 360N-CB-482, exceeds these minimum requirements. The minimum configuration for the CICS/DOS-STANDARD system satisfies the compilation requirements.

The following terminals, terminal control units, and programmable special features are supported by CICS. The user should be aware that many terminal and control unit special features are transparent to programming, and are therefore readily usable even though not specifically identified.

TERMINALS CONNECTED VIA NON-SWITCHED LINES USING BTAM

START STOP TRANSMISSION

- 1030 Data Collection System with:
  1031 Control Unit/Input Station and, optionally:
  1033 Printer
  1035 Badge Readers

- 1050 Data Communication System with:
  1051 Control Unit Model 1 or 2
  1052 Printer Keyboard with, optionally:
  1053 Printer Model 1
  1056 Card Reader

- 2260 Display Station Model 1 or 2 with:
  2848 Display Control Model 1, 2, or 3 with, optionally:
  Line Addressing (#4787), and/or
  1053 Printer Model 4

- 2265 Display Station with:
  2845 Display Control with, optionally:
  Line Addressing (#4801), and/or
  Tab (#7801), and/or
  1053 Printer Model 4

- 2740 Communication Terminal Model 1 with, optionally:
  Record Checking (#6114), and/or
  Station Control (#7479)

- 2740 Communication Terminal Model 2 with, optionally:
  Record Checking (#6114), and/or
  Buffer Receive (#1499)

- 2741 Communications Terminal
  For the CICS/DOS Systems, the Transmission Control Unit
  or ICA must be equipped with 2741 Break (#8055), or equivalent.

- 2760 Optical Image Unit attached to a
  2740 Communication Terminal Model 1 with:
  Record Checking (#6114)

- System/7
  5010 Processor Module Models A2-A16 with:
  Asynchronous Communications Control (#1610)

BINARY SYNCHRONOUS COMMUNICATION

- System/360 or System/370 via:
  Integrated Communications Attachment (Models 25 and 135)
  2701 Data Adapter Unit, or
  2703 Transmission Control

- System/360 Model 20 Processing Unit with:
  Binary Synchronous Communication Adapter (#2074), and
  EBCDIC Transmission Code (#9060), or
  ASCII Transmission Code (#9061), and, optionally:
  Station Selection (#7477)

- 2770 Data Communication System
  2772 Multipurpose Control Unit with:
  EBCDIC Transmission Code (#9761), or
  ASCII Transmission Code (#9762) and, optionally:
  WACK Response (#9936), and/or
  Buffer Expansion (#1490), and/or
  Conversational Mode (#1910), and/or
  Multi-point Data Link Control (#5010), and
  545 Output Punch, and/or
  1053 Printer, or
  2213 Printer, and/or
  2265 Display Station, and/or
  2502 Card Reader

- 2780 Data Transmission Terminal with:
  EBCDIC Transmission Code (#9761), or
  ASCII Transmission Code (#9762), or
  6-Bit Transcode (#9760) and, optionally:
  Multi-point Line Control (#5020)

- 2980 General Banking System
  2972 Terminal Control Unit Model 8 (RPQ858160), or
  2972 Terminal Control Unit Model 11 (RPQ8582311) with:
  2980 Teller Station Model 1 (RPQ835504), and/or
  2980 Administrative Station Model 2 (RPQ835505), and/or

2980 Teller Station Model 4 (RPQ858147) with, optionally:
Buffer Expansion (RPQ858165) for Models 1, 2, and 4, and/or
Auditor Key (RPQ858188) for 2980 Model 2

- 3270 Information Display System
  3271 Control Unit Model 1 or 2 with:
  3277 Display Station Model 1 or 2,
  3284 Printer Model 1 or 2,
  3286 Printer Model 1 or 2,
  3275 Display Station Model 1 or 2 with:
  Printer Adapter (#5550) for 3284 Printer Model 3
  and, optionally:
  ASCII Transmission Code (#1087)
  Keyboard Numeric Lock (#4690)
  Selector Pen (#6350)
  Audible Alarm (#1090)
  Security Keylock (#6340)
  Copy (#1550) for 3271 Control Unit

- System/3 Models 6 and 10
  5406 Processing Unit Models B2-B4, or
  5410 Processing Unit Models A2-A16, with:
  Binary Synchronous Communications Adapter (#2074) and,
  optionally:
  Station Selection (#7477)

- 1130 Computing System with:
  Synchronous Communications Adapter (#7690)

TERMINALS CONNECTED VIA SWITCHED LINES USING BTAM

START STOP TRANSMISSION

- 1050 Data Communication System with:
  1051 Control Unit Model 1 or 2
  1052 Printer Keyboard with, optionally:
  1053 Printer Model 1
  1056 Card Reader

- 2740 Communication Terminal Model 1 with:
  Dial-Up (#3255) and, optionally:
  Record Checking (#6114)

- 2741 Communications Terminal with:
  Dial-Up (#3255)
  For the CICS/DOS Systems, the Transmission Control Unit or ICA
  must be equipped with 2741 Break (#8055), or equivalent.

- 2760 Optical Image Unit attached to a
  2740 Communication Terminal Model 1 with:
  Dial-Up (#3255), and
  Record Checking (#6114)

- System/7
  5010 Processor Module Models A2-A16 with:
  Asynchronous Communications Control (#1610)
  Autocall (#1310) on 2702 Transmission Control, or
  Autocall (#1340) on 2703 Transmission Control

- TWX Common Carrier Teletypewriter Exchange Terminal Station
  (Model 33/35) eight-level code at 110 bps on common carrier
  switched 150-baud networks

BINARY SYNCHRONOUS COMMUNICATION

- System/360 or System/370 via:
  Integrated Communications Attachment (Models 25 and 135)
  2701 Data Adapter Unit, or
  2703 Transmission Control

- System/360 Model 20 Processing Unit with:
  Binary Synchronous Communication Adapter (#2074), and
  EBCDIC Transmission Code (#9060), or
  ASCII Transmission Code (#9061) and, optionally:
  Automatic Calling (#1315)

- 2770 Data Communication System
  2772 Multipurpose Control Unit, with:
  EBCDIC Transmission Code (#9761), or
  ASCII Transmission Code (#9762) and, optionally:
  WACK Response (#9936), and/or
  Buffer Expansion (#1490), and/or
  Conversational Mode (#1910), and/or
  Automatic Answering (#1340), and/or
  Identification (#4610), or
  Security Identification (#6310), and
  545 Output Punch, and/or
  1053 Printer, or
  2213 Printer, and/or
  2265 Display Station, and/or
  2502 Card Reader

- 2780 Data Transmission Terminal with:
  EBCDIC Transmission Code (#9761), or
  ASCII Transmission Code (#9762), or
  6-Bit Transcode (#9760) and, optionally:
  Automatic Answering (#1340)

- 3735 Programmable Buffered Terminal (CICS/OS only) with:
  EBCDIC Transmission Code (#9761), or
  ASCII Transmission Code (#9762)

- 1130 Computing System with:
  Synchronous Communications Adapter (#7690)

- System/3 Models 6 and 10
  5406 Processing Unit Models B2-B4, or
  5410 Processing Unit Models A2-A16, with:
  Binary Synchronous Communications Adapter (#2074) with, optionally:
  Automatic Calling (#1315)

TERMINALS CONNECTED VIA LOCAL ATTACHMENT USING BTAM

- 2260 Display Station Model 1 or 2 with:
  2848 Display Control Model 1, 2, 3, 21, or 22 with, optionally:
  Line Addressing (#4787), and/or
  1053 Printer Model 4

- 3270 Information Display System
  3272 Control Unit Model 1 or 2 with:
  3277 Display Station Model 1 or 2, and/or
  3284 Printer Model 1 or 2, and/or
  3286 Printer Model 1 or 2, and, optionally:
  Keyboard Numeric Lock (#4690)
  Selector Pen (#6350)
  Audible Alarm (#1090)
  Security Keylock (#6340)

- 7770 Audio Response Unit Model 3
  Touch-Tone* telephone, or equivalent equipment, and the IBM 2721 Portable Audio Terminal are supported through the 7770 Audio Response Unit Model 3.

TERMINALS SUPPORTED UNDER CICS/OS USING TCAM

The following terminals are supported by CICS/OS using TCAM. Only those terminal features supported by both CICS/OS and TCAM are applicable for use by CICS application programs which are associated with terminals attached to TCAM. For information concerning terminals supported by TCAM, see the _OS TCAM Programmer's Guide and Reference Manual_ (GC30-2024).

Switched and Non-Switched	Non-Switched	Local Attachment
1050	2260	2260
2740 Model 1	2265	3270
2741	2740 Model 2	7770
System/370	3270	
2770		
2780		
System/3		
TWX Model 33/35		

Note: The user should be aware that TCAM supports some terminals and Terminal Control units not supported by CICS, and conversely.

*Trademark of the American Telephone & Telegraph Co.

STORAGE ESTIMATES AND CONSIDERATIONS

CICS is distributed in the form of Assembler language source statements (80-column card images). System generation requires the necessary online storage devices and a capacity to assemble a 2400-statement program.

The distributed and generated libraries require up to 2.5 million bytes of direct access storage, depending on system options generated and blocking factors chosen by the user (where applicable).

The system configuration required to use CICS as a data base/data communication interface is largely determined by the scope of the environment to be supported and by the nature of the user's application(s). CICS is designed in a modular fashion with many system generation options so that the user can tailor the system to meet his requirements with maximum economy of main storage.

It is possible to generate a minimum CICS environment for an application with several start/stop terminals of a single type. Although such an environment may satisfy the needs of some users, it would not represent a practical operating environment for the majority of CICS users who have considerably greater system requirements and plans. The minimum CICS environment does represent a possible starting point for the majority of users for initial development, testing, and implementation of a pilot application with several terminals. The following minimum CICS main storage requirements do not include the main storage occupied by the operating system. They do address both the static and dynamic storage requirements discussed below.

CICS/DOS-ENTRY	30K
CICS/DOS-STANDARD	44K
CICS/OS-STANDARD V2	64K

The Storage Estimates worksheet provided later in this section should be completed to determine the actual main storage required for the user's application.

MAIN STORAGE REQUIREMENTS

The CICS partition/region is comprised of two types of main storage:

1. Static storage
2. Dynamic storage

CICS static storage is located in the "high" end of the partition/region and is allocated during system initialization. It contains the CICS management and service programs, CICS control tables, access methods, and resident user-written application programs.

CICS dynamic storage occupies the remainder of the partition/region and contains control areas, input/output areas, work areas, application programs to be processed, and the storage cushion.

The dynamic storage area (containing application programs, control areas, input/output areas, and work areas) must be large enough to provide the throughput and response as required by the user for his application within his environment.

The CICS/DOS-ENTRY dynamic storage area must accommodate the System Initialization program (4K) plus the largest application program to be stored in the Real-Time Pre-Located Program Library. The CICS system service programs do not exceed 4K. Therefore, if these optional programs are to be used, and if the user's largest application processing program does not exceed this figure, the dynamic storage area may be as low as 8K, providing the user's other storage requirements are also met.

STATIC STORAGE

CICS static storage for a given system must be large enough to contain the programs, tables, control areas, and access methods required by that system. The required size of static storage can be estimated as the sum of the following:

1. Required CICS management programs (Task Control, Storage Control, Program Control, and Terminal Control)
2. Optional CICS management programs (Interval Control, Program Interrupt Control, File Control, Transient Data Control, Temporary Storage Control, Trace Control, and Dump Control)
3. Required CICS tables and control areas (Program Control Table, Processing Program Table, Terminal Control Table, CICS/DOSE Rollout Table, and Common System Area)
4. Optional CICS tables (File Control Table, Destination Control Table, Temporary Storage Table, and Trace Table)
5. Operating system access methods (DAM is always required)
6. Resident application programs

The following tables contain estimates of the number of bytes of static storage required for the above. However, the user must make his own estimates concerning items 5 and 6.

<u>Note:</u> If TCAM is used under CICS/OS, the size of the user-written TCAM Message Control program (which resides in another partition/region) should be taken into consideration when computing storage estimates.

Table 1. Task Control Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Routines	DFHSG PROGRAM=KCP	2800	2100	2100
Trace Support	DFHSG TYPE=INITIAL,TRACE=YES	+25	+25	+25
Operator Security	DFHSG PROGRAM=KCP, OPSECUR=YES,	+40	+40	+40
Enqueue/Dequeue	ENQUEUE=YES,	+700	+775	+775
Consecutive Dispatch Control	CONSEC=YES,	+12	N.A.	N.A.
Stall Control	STALCTL=YES	+180	+190	+190
Interval Control Basic Support (note 1)	DFHSG TYPE=INITIAL, TIMECTL=YES	+60	+60	+100
Time of Day Support (note 2)	DFHSG PROGRAM=KCP, ICPTIME=YES	+0	+0	+0
Runaway Task Support	DFHSG PROGRAM=KCP, RUNAWAY=YES	+60	+60	+80
Time-Dependent Event Basic Routines	DFHSG PROGRAM=KCP, ICPAUTO=YES|ICPSYNC=YES	+170	+170	+170
Automatic Task Initiation (note 3)	DFHSG PROGRAM=KCP, ICPAUTO=YES	+50	+50	+50
Task Synchronization (note 3)	DFHSG PROGRAM=KCP, ICPSYNC=YES	+0	+0	+0

Note 1: Interval Control support code is required in Task Control only if the Interval Control program (DFHICP) of CICS Time Management is being used.

Note 2: All code necessary to support the time-of-day feature is contained in the basic routines generated to support Interval Control.

Note 3: All code necessary to support Time-Dependent Task Synchronization and Automatic Task Initiation is contained in the basic routines generated to support time-dependent events.

Table 2. Storage Control Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Routines	DFHSG PROGRAM=SCP	1660	1625	1600
Trace Support	DFHSG TYPE=INITIAL, TRACE=YES	+80	+80	+80

Table 3. Program Control Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Routines (Including DTF/DCB)	DFHSG PROGRAM=PCP	1200	1400	1400
Load Option Support	PCPLOAD=YES	+700	+300	+300
Trace Support	DFHSG TYPE=INITIAL,TRACE=YES	+25	+25	+25
High-Level Language Trace Support	DFHSG PROGRAM=PCP, HLLTR=YES	+200	+200	+200
High-Level Language Support Basic Required Routines	LANG=COBOL|PL/I	+150	+150	+150
PL/I Support	LANG=PL/I	+220	+230	+230
ANS COBOL Support	LANG=COBOL	+770	+790	+800
ATP Support	DFHSG TYPE=INITIAL,ATP=YES	N.A.	+30	+30
DL/I Support	DFHSG TYPE=INITIAL,DL1=YES	N.A.	N.A.	+30

Table 4 (Part 1 of 3). Terminal Control Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Routines	DFHSG PROGRAM=TCP	2000	2000	2400
TCAM Support	ACCMETH=TCAM	N.A.	N.A.	+2000
Programmable Device Support	BTAMDEV=any programmable device	----	----	----
Basic		+100	+100	+100
Bisync (note 1) Switched	BTAMDEV=1130D|2020D|SYS/3D|S/360D|S/370D	+370	+370	+370
Non-Switched	BTAMDEV=1130|2020|SYS/3|S/360|S/370	+320	+320	+320
Automatic Transaction Initiation	AUTOTRN=YES	+140	+140	+120
Automatic Answerback	ANSWRBK=AUTOMATC	+140	+140	+140
Expanded ID Verification	ANSWRBK=EXIDVER	N.A.	N.A.	+92
Translate Tables (note 2)				
2740, 2741, System/7	BTAMDEV=2740|2741E|SYS/7	+520	+520	+520
2741 Correspondence	BTAMDEV=2741C	+520	+520	+520
Start/Stop Support				
Non-Switched	BTAMDEV=any non-switched, start/stop device	+300	+300	+300
Autopoll Support (note 3)	FEATURE=AUTOPOLL	+530	+530	+150
Wraplist Support (note 4)	WRAPLST=YES	+150	+150	+150
Switched	FEATURE=AUTOANSW and BTAMDEV=any switched start/stop device	+370	+370	+350
Bisynchronous Basic Support	BTAMDEV=any bisync device	+3200	+3200	+3000
Non-Switched	BTAMDEV=any non-switched bisync device	+460	+460	+420
Switched	BTAMDEV=any switched bisync device	+1000	+1000	+900
Bisynchronous Translate Tables	BSCODE=TRANSCODE	+520	+520	+520
	BSCODE=ASCII	+520	+520	+520
	BSCODE=EBCDIC	0	0	0
SAM Basic Support	DEVICE=	+350	+350	+230
Reader/Printer	DEVICE=(2540, 1403...)	+100	+100	N.A.
Tape and Disk	DEVICE=TAPE|DISK	+80	+80	N.A.
1030 Support	BTAMDEV=1030	+1050	+1050	+980
1050 Support	BTAMDEV=1050|1050D			
Translate Table	BTAMDEV=1050|1050D	+520	+520	+520
Switched Support	BTAMDEV=1050D	+600	+600	+520
Non-Switched Support	BTAMDEV=1050	+600	+600	+575

Table 4 (Part 2 of 3). Terminal Control Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES DOSE	DOSS	OS
2260-2265 Support	BTAMDEV=2260	+1300	+1300	+1250
2265	BTAMDEV=2265	+30	+30	N.A.
1053 Support	BTAMDEV=1053	+70	+70	+60
Local 2260	BTAMDEV=L2260	+700	+700	+480
Lock Option	LOCKF=YES	+50	+50	+60
1053 Support	BTAMDEV=1053	+70	+70	+60
2740 Support	BTAMDEV=2740|2740D| 2740-2	+20	+20	+20
Switched	BTAMDEV=2740D	+600	+600	+530
Non-Switched	BTAMDEV=(2740,2740-2)	+580	+580	+530
2740 Model 2	BTAMDEV=2740-2	+70	+70	+70
2740 Model 2 Buffer Receive	FEATURE=BUFFRECV	+180	+180	+180
2760 OIU	BTAMDEV=2760	+100	+100	+90
2741 Support	BTAMDEV=2741C|2741E| 2741DC|2741DE	+44	+44	+44
Switched	BTAMDEV=2741DC|2741DE	+450	+450	+410
Non-Switched	BTAMDEV=2741C|2741E	+280	+280	+260
2770 Support	BTAMDEV=2770|2770D			
Switched	BTAMDEV=2770D	+370	+370	+370
Non-Switched	BTAMDEV=2770	+320	+320	+320
2780 Support	BTAMDEV=2780|2780D			
Switched	BTAMDEV=2780D	+350	+350	+350
Non-Switched	BTAMDEV=2780	+300	+300	+300
2980 Support (Common)	BTAMDEV=2980/1|2980/2| 2980/4|2980/2A	+2300	+2300	+2300
Model 1	BTAMDEV=2980/1	+330	+330	+330
Model 2	BTAMDEV=2980/2|2980/2A	+330	+330	+330
Model 4	BTAMDEV=2980/4	+350	+350	+350
3270 Support (Common) (note 5)	BTAMDEV=3277|L3277|3284| L3284|3275|3286| L3286			
	or COMPAT=FULLBUF|FORMAT	+150	+150	+150
3270 Native Mode	BTAMDEV=3275|3277|L3277	+350	+350	+120
Local 3270	BTAMDEV=L3277	+1400	+1400	+1400
Remote 3270	BTAMDEV=3275|3277	+2600	+2600	+2600
Uppercase Translate	UCTRAN=ASCII|EBCDIC			
EBCDIC	UCTRAN=EBCDIC	+270	+270	+270
ASCII	UCTRAN=ASCII	+280	+280	+280

Table 4 (Part 3 of 3). Terminal Control Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
3270/2260 Compatibility	COMPAT=FULLBUF|FORMAT	+4000	+4000	+4000
FULLBUF and 480-character mapping	COMPAT=FULLBUF & FMT2260=12X40 & FMT3270=12X40	+220	+220	+220
Tab Support	TAB=YES	+60	+60	+60
3284/3286 Support	BTAMDEV=3284|L3284|3286|L3286	+200	+200	+200
3735 (dial) support	BTAMDEV=3735D	N.A.	N.A.	+1500
7770 Support	BTAMDEV=7770	+670	+670	+650
Data Conversion ABB'	CONVTAB=ABB	+330	+330	+330
Data Conversion ABC	CONVTAB=ABC	+280	+280	+280
System/7 Support	BTAMDEV=SYS/7|SYS/7D	+250	+250	+250
Switched	BTAMDEV=SYS/7D	+570	+570	+570
Non-Switched	BTAMDEV=SYS/7	+800	+770	+770
Pseudo-binary tran code	FEATURE=PSEUDOBIN	+650	+650	+650
TWX Support	BTAMDEV=TWX	+1000	+1000	+1000
TCAM 3270 Support	TCM3270=YES	N.A.	N.A.	+525
TCAM 7770 Support	TCM7770=YES	N.A.	N.A.	+210

Note 1: Since the code supporting programmable binary synchronous devices for both switched and non-switched lines is included in 2770 support, these storage estimates can be ignored if the corresponding 2770 support (switched and/or non-switched) is specified (see part 2 of this table).

Note 2: Those translate tables not identified here are included in the storage estimates for the specific device to which they apply.

Note 3: Autopoll support is applicable only to the 1030, 1050, and 2740, and multipoint binary synchronous devices.

Note 4: Wraplist support is applicable only to the 1030, 1050, 2740, and 2260 remote.

Note 5: For storage estimates concerning 3270 Basic Mapping support (BMS) see the section "3270 Basic Mapping Support".

Table 5. Interval Contrcl Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Routines	DFHSG PROGRAM=ICP	550	560	350
Runaway Task Control	RUNAWAY=YES	+100	+100	+225
Time-of-day Support	ICPTIME=YES	+75	+75	+75
Trace Support	DFHSG TYPE=INITIAL,TRACE=YES	+25	+25	+25
Time-Dependent Events	DFHSG PROGRAM=ICP,			
Basic Support	ICPAUTO=YES|ICPSYNC=YES	+720	+720	+720
Automatic Task Initiation	ICPAUTO=YES	+900	+900	+900
Task Synchronization	ICPSYNC=YES	+250	+250	+250
Dummy Program (see note)	DUMMY=YES	60	60	60

Note: If a dummy Interval Control Program is generated, this storage estimate is the total storage requirement for the Interval Control Program.

Table 6. Program Interrupt Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Routine	DFHSG PROGRAM=PIP	284	284	500

Table 7. File Control Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES DOSE	 DOSS	 OS
Required Routines	DFHSG PROGRAM=FCP	525	580	625
Dynamic Open/Close/Locate	FILSERV=LOCATE	+110	+110	+110
Add & Update Support	FILSERV=DAUPD|DAADD|ISUPD|ISADD	+40	+40	+40
Update Support	FILSERV=DAUPD|ISUPD	+80	+80	+95
Add Support	FILSERV=DAADD|ISADD	+80	+80	+60
Read & Update Support	FILSERV=INDA|DAUPD|INIS|ISUPD	+85	+85	+90
DAM Basic Support	FILSERV=INDA|DAUPD|DAADD	+220	+225	+170
DAM Read	FILSERV=INDA	+8	+8	+15
DAM Update	FILSERV=DAUPD	+105	+105	+25
DAM Add	FILSERV=DAADD	+340	+340	+45
ISAM Basic Support	FILSERV=INIS|ISUPD|ISADD	+140	+190	+190
ISAM Read	FILSERV=INIS	+42	+46	+25
ISAM Update	FILSERV=ISUPD	+130	+130	+375
ISAM Add	FILSERV=ISADD	+162	+168	+8
ISAM Variable Length Records	FILSERV=IVBR	+100	+100	N.A.
DAM Blocking	FILSERV=DABLKNG	+150	+150	+150
Indirect Accessing	FILSERV=INDIRACC	+155	+155	+160
Browse Basic Support	FILSERV=IBROWSE|DBROWSE	+330	+320	+475
ISAM Browse	FILSERV=IBROWSE	+220	+220	+175
DAM Browse	FILSERV=DBROWSE	+375	+375	+300
Exclusive Control	FILSERV=EXCTL	+170	+170	+50
Variable Length Records (DAM or ISAM)	FILSERV=VLR	+140	+140	+160
Input Segmenting	FILSERV=INSEG	+475	+475	+475
Output Segmenting	FILSERV=OUTSEG	+240	+240	+240
DAM Relative Track Reference	FILSERV=HEXAD|DECAD	+110	+110	+150
DAM Actual Track Reference	FILSERV=ACTAD	+95	+85	+100
Consecutive Task Dispatching Feature	DISPLIM=YES	+30	N.A.	N.A.
Dummy Program (note 1)	DUMMY=YES	50	50	50
DL/I Support (note 2)	DFHSG TYPE=INITIAL,DL1=YES	N.A.	N.A.	+Si
Trace Support	DFHSG TYPE=INITIAL,TRACE=YES	+25	+25	+25

Note 1: If the dummy File Control program is generated, the other features are not needed, and vice versa.

Note 2: Si = +14,000 bytes. In addition, storage is also required for an IMS batch system (approximately 2,000 bytes) and the routines and blocks required to handle the CICS/OS user's DL/I requests. For more information concerning these IMS-DL/I requirements, see the IMS/360 Version 2 System/Application Design Guide (SH20-0910).

After the total IMS-DL/I storage requirement has been determined, enter this figure in the appropriate place on the the Storage Estimates Worksheet. This figure is also used to determine the OSCOR requirement at CICS/OS system generation.

Table 8. Transient Data Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Routines (Includes DTF/DCB)	DFHSG PROGRAM=TDP	600	600	300
Support Extrapartition Basic Routines	EXTRA=ACQUISITION| DISPOSITION	+80	+80	+110
Extrapartition Data Acquisition	EXTRA=ACQUISITION	+60	+60	+10
Extrapartition Data Disposition	EXTRA=DISPOSITION	+10	+10	+10
Support Intrapartition	INTRA=YES	+2200	+2200	+2800
Automatic Transaction Initiation	INTRA=TRANSINIT	+160	+160	+160
Dummy Program (see note)	DUMMY=YES	75	75	75
Trace Support	DFHSG TYPE=INITIAL,TRACE=YES	+25	+25	+25

Note: If the dummy Transient Data Control program is generated, the other features are not needed, and vice versa.

Table 9. Temporary Storage Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Routines	DFHSG PROGRAM=TSP	1000	1000	1000
Trace Support	DFHSG TYPE=INITIAL,TRACE=YES	+25	+25	+25
Dummy Program (see note)	DUMMY=YES	45	45	45

Note: If a dummy Temporary Storage Program is generated, the other features are not needed, and vice versa.

Table 10. Trace Control Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Routines	DFHSG TYPE=INITIAL,TRACE=YES	900	900	900

Table 11. Dump Control Program - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Routines Disk, including DTF/DCB and Access Method (see note)	DFHSG DEVICE=2311|2314	2050	2060	1200
Required Routines Tape, including DTF/DCB and Access Method (note 1)	DEVICE=TAPE	1400	1410	1200
List CICS Modules and Tables	CICSDMP=YES	+280	+220	+220
Dummy Program (Note 2)		25	25	25

features are not needed, and vice versa.

Note 1: If the CICS/DOS "dump to disk" facility is elected by the user, the "dump to tape" facility is not needed, and vice versa. In the CICS/OS system, there is no difference between these facilities. Refer to the publication OS/360 Storage Estimates (GC28-6551) for the size of the access method module(s) associated with CICS/OS sequential data sets.

Note 2: If a dummy Dump Control Program is generated, the other features are not needed, and vice versa.

Table 12. Asynchronous Transaction Processing program (ATP) - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Asynchronous Transaction Control (DFHATP)	DFHSG PROGRAM=ATP	----	2700	2700
Asynchronous Transaction Input Processor (DFHRDR)	DFHSG PROGRAM=ATP	----	2500	2500
Asynchronous Transaction Output Processor (DFHWTR)	DFHSG PROGRAM=ATP	----	3700	3700
Asynchronous Queue Purge (DFHAQP)	DFHSG PROGRAM=ATP	---	150	150

Note: The Asynchronous Transaction Processing (ATP) routines are nonresident routines that are brought into dynamic main storage only under the following conditions:

DFHATP - in main storage any time batches are being processed.

DFHRDR - in main storage only when batches are read in from terminals.

DFHWTR - in main storage only when output from batch processing is transmitted to a terminal.

DFHAQP - in main storage only when the direct access reservoirs containing input/output are being purged.

Table 13. Program Control Table (PCT) - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Each PCT Entry	DFHPCT TYPE=ENTRY	32	32	32

Note: PCT size = 24 + (N * 32), where N is the number of table entries (transaction codes) and 32 is the size of each entry. As many as ten transaction code entries can be made for CICS service programs.

Table 14. Processing Program Table (PPT) - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Each PPT Entry	DFHPPT TYPE=ENTRY, PGMLANG=ASSEMBLER	28	28	32
High-Level Language Extension	DFHPPT TYPE=ENTRY, PGMLANG=COBOL|PL/I	+12	+12	N.A.

Note: PPT size for CICS/DOS = 24 + (Na * 28) + (Nh * 40), where Na is the number of application programs written in Assembler language and Nh is the number of application programs written in a high-level language (ANS COBOL or PL/I). The number of application programs can include up to 15 CICS service programs.

PPT size for CICS/OS = 24 + (Np * 32), where Np is the number of application programs, regardless of the language used.

Table 15. Terminal Control Table (TCT) - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Operating System Control Block (DTF/DCB) (note 1)	DFHTCT TYPE=SDSCI			
	DEVICE=2540	50	50	88
	DEVICE=1403	54	54	88
	DEVICE=TAPE	48	48	88
	DEVICE=DISK	152	152	88
	DEVICE=(2741C|2741E)	152	152	36
Constants for each Line Entry (note 2)	DFHTCT TYPE=LINE	84	84	84
Pool Addr for Dial Terminals	POOLADR=name	+4	+4	+4
Bisynchronous	DEVICE=any bisync device	+20	+20	+20
Constants for each Terminal Entry (note 3)	DFHTCT TYPE=TERMINAL	68	68	68
Last Multidropped Terminal Only				
Openlist	LASTTRM=LINE	+8	+8	+8
Autopoll Line Feature	LASTTRM=LINE	+12	+12	+12
User-specified DFTRMLST for addressing	TRMADDR=addr	+5	+5	+5
Bisync	CLASS=BISYNC or DEVICE=any bisync device	+16	+16	+16
3270 Display	BTAMDEV=3270	+20	+20	+20
3270 Compatibility	COMPAT=(no. of char,no. of lines,device type, model no.)	+12	+12	+12
3735 (Dial-up)	BTAMDEV=3735D	N.A.	+1	+1
DFTRMLST Requirements for polling (note 4)				

Note 1: For other BTAM or TCAM devices, see the appropriate operating system storage estimates.

Two DFHTCT TYPE=SDSCI macro instructions must be included for each sequential terminal; one for the sequential input data set and one for the sequential output data set. This input/output data set combination simulates the input/output functions of a terminal. One DFHTCT TYPE=LINE macro instruction must be coded for this I/O combination.

One DFHTCT TYPE=SDSCI macro instruction must be coded for each BTAM line group, where a line group is a group of communication lines that meet the following operational requirements:

1. All lines in the group are attached to the channel through the same type of telecommunications control unit; for example, a 2701. (This is true only for the CICS/DOS systems.)

2. The line connection between the control unit and the remote devices is of the same type; for example, a switched network.

3. All devices within the line group have the same line features and operating characteristics; for example, autopoll.

Two DFHTCT TYPE=SDSCI macro instructions must be coded for each sequential TCAM process queue; one for the TCAM input queue and one for the TCAM output queue. At least two DFHTCT TYPE=LINE macro instructions must be coded for this I/O combination; one for the TCAM input queue and one for the TCAM output queue.

Note 2: One DFHTCT TYPE=LINE macro instruction must be included for each line or TCAM process queue.

Note 3: One DFHTCT TYPE=TERMINAL macro instruction must be included for each terminal or task (if TCAM is the access method chosen and POOL=YES has been specified).

Note 4: See the appropriate operating system BTAM manual for storage estimates concerning the use of this macro instruction.

Note 5: The size of the TCT may be calculated as follows:

$$18 + 4(Ncb + Nl) + (St1 + St2 + \ldots + Stn) + (Sl1 + Sl2 + \ldots + Sln) + (Scb1 + Scb2 + \ldots + Scbn) + D,$$ where:

Ncb = number of control blocks (that is, DTF/DCB's)
Nl = number of line entries (that is, TYPE=LINE)
St1 = size of the first terminal entry as determined from Table 15
St2 = size of the second terminal entry
Stn = size of the nth terminal entry
Sl1 = size of the first line entry
Sl2 = size of the second line entry
Sln = size of the nth line entry
Scb1 = size of the first control block (that is, DTF/DCB)
Scb2 = size of the second control block
Scbn = size of the nth control block
D = size of the terminal list as determined from the publication referenced in note 4

Table 16. Common System Area (CSA) - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Required Constants	DFHSG PROGRAM=CSA,	1840	1850	1850
Common Work Area	WRKAREA=number	+N	+N	+N

Note 1: The storage estimates for required constants include storage estimates for (1) the CSA, (2) the Task Control Areas for the Terminal Control program and Task Control program, and (3) in the case of the CICS/DOS systems, the Write to Console routine.

Note 2: Size of the Common Work Area is user-defined. (N = number of bytes.)

Table 17. File Control Table (FCT) - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Basic Constants	DFHFCT TYPE=DATASET	48	48	48
Indirect Access Extension	DFHFCT TYPE=INDACC	+21	+21	+21
Segment Extension	DFHFCT TYPE=SEGDEF	+Sx	+Sx	+Sx
(note 1)	TYPE=SEGSET	+14S	+14S	+14S
DAM Operating System Control Block (note 2)	DFHFCT TYPE=DATASET, ACCMETH=BDAM	250	250	+104
ISAM Operating System Control Block	DFHFCT TYPE=DATASET, ACCMETH=ISAM	---	---	252
Add Only	SERVREQ=NEWREC	530	530	N.A.
Retrieve Only	SERVREQ=GET	300	300	N.A.
Add/Retrieve (note 3)	SERVREQ=GET,NEWREC	550	550	N.A.
ISAM Resident Index Area (note 5)	DFHFCT TYPE=DATASET, INDAREA=symbol, INDSIZE=Si	+Si	+Si	+Si
ISAM Register Save Area	DFHFCT TYPE=DATASET, ACCMETH=ISAM	N.A.	100	N.A.
Prime Data in Main Storage Area (MSWA for OS)	DFHFCT TYPE=DATASET, ACCMETH=ISAM, IOSIZE=n	+n	+n	+n

Note 1: Sx = 8 + 2D, where D is the number of segments per record (TYPE=SEGDEF); S is the number of segment sets (TYPE=SEGSET).

Note 2: This is an average size.

Note 3: Add the key length to each of these values.

Note 4: Size of FCT = 8 + (4 * Nd) + Se1 + Se2 + ... + SeN), where Nd is the number of data sets (files) defined in the FCT and Se is the size of each table entry as determined from Table 16. Each entry in the table consists of the basic constants (DFHFCT TYPE=DATASET) and either an ISAM or DAM operating system control block (DTF/DCB). For every data set that is to be a cross index to another data set (indirect accessing), the Indirect Access Extension is required. For every data set that contains segmented records, the Segment Extension is required. For every ISAM data set that is to have a resident cylinder index, the ISAM Resident Index Area is required. For every ISAM data set with the "prime data in main storage" option, the Prime Data in Main Storage Area is required. In the CICS/DOS-STANDARD system, all ISAM data sets require the ISAM Register Save Area.

Note 5: Si = size of the main storage area reserved for processing of the highest level index, as specified in the INDSIZE operand.

Table 18. Destination Control Table (DCT) - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Intrapartition Destination Basic Constants	DFHDCT TYPE=INTRA	20	20	20
Automatic Transaction Initiation	DFHDCT TYPE=INTRA,TRANSID='xxxx'	+4	+4	+4
Extrapartition Destination Basic Constants	DFHDCT TYPE=EXTRA	12	12	12
Indirect Access Basic Constants	DFHDCT TYPE=INDIRECT	+12	+12	+12
Operating System Control Block (DTF/DCB)	DFHDCT TYPE=SDSCI			
Automatic Open (note 1)	OPEN=INITIAL,BUFNO=n	+N	+N	N.A.
Printer	DEVICE=1403|1404|1443|1445	60	60	96
Disk	DEVICE=2311|2314			
Fixed Input Data Set	RECFORM=FIXBLK|FIXUNB, TYPEFLE=INPUT	140	140	96
Fixed Output Data Set	RECFORM=FIXBLK|FIXUNB, TYPEFLE=OUTPUT	172	172	96
Variable Output Data Set	RECFORM=VARBLK|VARUNB, TYPEFLE=OUTPUT	180	180	96
Tape	DEVICE=TAPE			
Standard Labels	FILABL=STD	16	16	96
Fixed Records	RECFORM=FIXBLK|FIXUNB	100	100	96
Variable Records	RECFORM=VARBLK|VARUNB	112	112	96

Note 1: N = n * Bl, where Bl is the buffer length. Note that buffers are obtained by the OS access methods at OPEN. The user must allow for this storage when considering his dynamic storage requirements.

Note 2: DCT size = 14 + (Se1 + Se2 + ... + SeN) + AM, where Se is the size of each entry as determined from Table 18, and AM is the total number of bytes required for all access methods when using the CICS/DOS systems. For CICS/OS, the access methods are not part of the DCT.

There must be one entry in the DCT for every symbolic destination that is to be referenced by the Transient Data Control program. Each entry is identified as either an intrapartition, extrapartition, or indirect destination. In addition, every physical extrapartition data set used by the Transient Data Control program must be described through use of an operating system control block (DTF/DCB). Note that more than one symbolic destination can be routed to one physical data set, thus requiring only one operating system control block.

One access method (logic) module is required to support each defined category of data sets in the CICS/DOS systems:

CATEGORY	SIZE
Printer - fixed length	104
Printer - variable length	168
Tape - fixed length	928
Tape - variable length	1088
Disk - fixed input	840
Disk - fixed output	816
Disk - variable input	1024
Disk - variable output	1233

Table 19. Temporary Storage Table (TST) - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Basic Entry		10	10	10

Note: Size cf Temporary Storage Table = 36 + (10 * Nb) where Nb is the number of blocks on the Temporary Storage data set. Nb is calculated as follows: Nb = Bt * Nt, where Bt is the number of blocks per track and Nt is the number of tracks allocated to the data set. The number of blocks per track (Bt) is device dependent.

Table 20. Trace Table (TRT) - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Basic entry	DFHSIT TYPE=CSECT, TRT=decimal value	16	16	16

Note: Size of Trace Table = 56 + (16 * Ne), where Ne is the number of entries in the Trace Table as specified in the System Initialization Table (DFHSIT TRT=decimal value).

Table 21. Rollout Table (ROT) - storage estimates

FEATURE	GENERATION OPERAND	NUMBER OF BYTES		
		DOSE	DOSS	OS
Basic entry		8	N.A.	N.A.

Note: Size of Rollout Table = 36 + (8 * MXT), where MXT is the maximum number of tasks allowed to process concurrently within the system. This value is provided by the user at System Initialization through use of the System Initialization Table.

DYNAMIC STORAGE - CICS/DOS-ENTRY SYSTEM

During initiation and execution of a transaction (task), the Task Control Area (TCA) and other storage areas required by the transaction are dynamically allocated by CICS. For the CICS/DOS-ENTRY system, dynamic storage is that part of main storage within the CICS partition that is available after the CICS nucleus and resident application programs have been loaded by System Initialization. This dynamic area is divided into two unique areas called Subpool 1 and Subpool 0. The storage organization for the CICS/DOS-ENTRY system is illustrated in Figure 16.

Subpool 1 contains transaction-oriented data that is rolled out to auxiliary storage whenever the transaction is waiting on an event of extended duration, such as a terminal read. Subpool 1 also contains the application program associated with the transaction unless that program was made permanently resident during System Initialization. If permanently resident, the program is located in the Resident Application Program (RAP) area. File Management I/O areas and CLASS=USER storage are examples of data allocated dynamically in Subpool 1.

Subpool 0 contains transaction-oriented data that, because of its nature, cannot be rolled out with the transaction but must remain in main storage for the life of the transaction or until released by the transaction. This dynamic data may be associated with a great number of transactions, all but one of which are in a rolled-out status. Terminal I/O Areas and Task Extension Areas (TXA's) are examples of data found in Subpool 0.

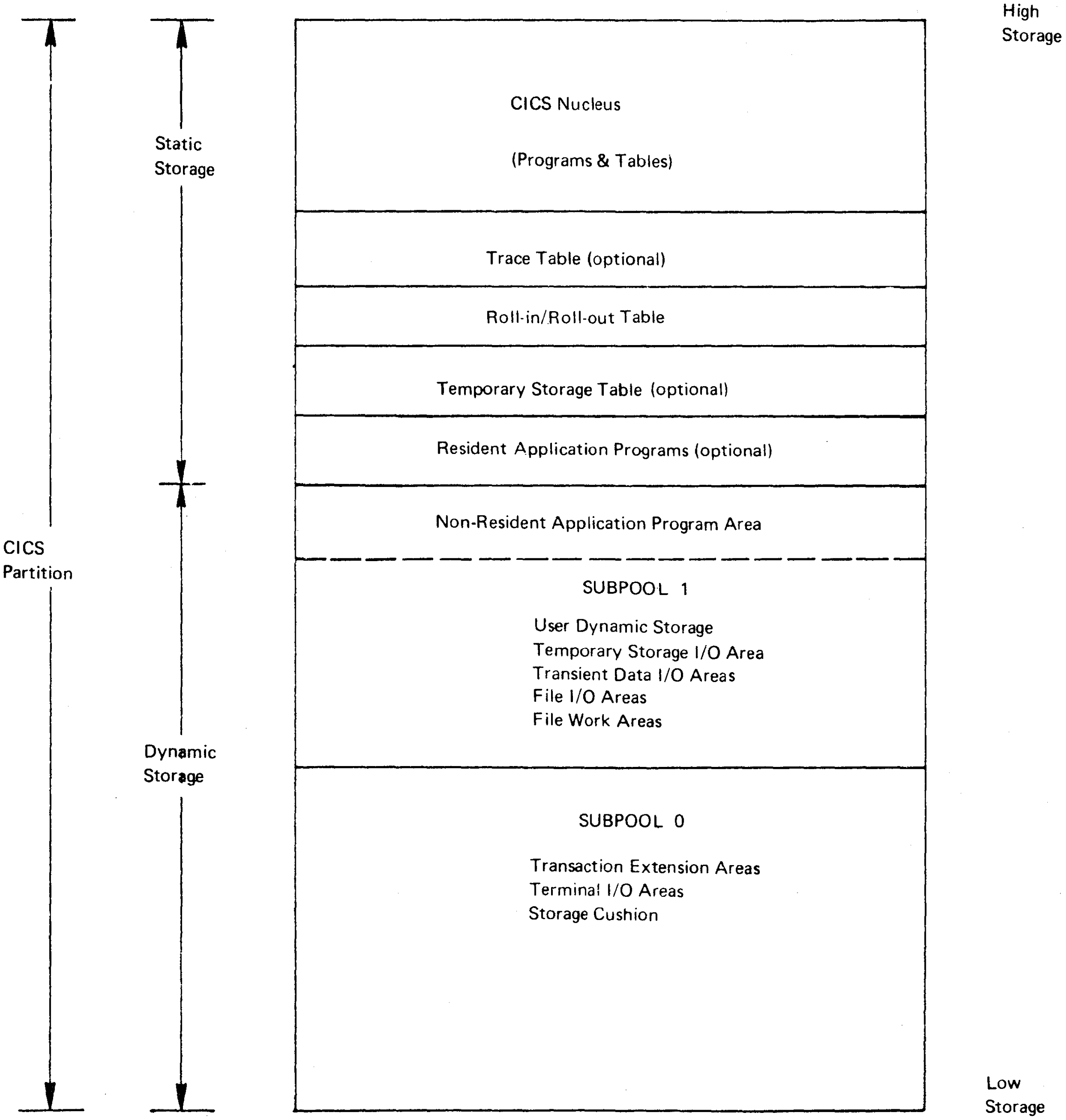


Figure 16. Storage organization for the CICS/DOS-ENTRY system

<u>Subpool 1</u>

The size of Subpool 1 is specified during System Initialization and does not change until the next System Initialization. The actual dynamic area available within this subpool varies from transaction to transaction and is equal to the specified size of the subpool less the size of the active application program (if not permanently resident), less the size of the TCA (256 bytes), and less the size of the TWA for the active task.

For example, assume that Subpool 1 has a size of 10,000 bytes and that application program ABCD has a size of 3,000 bytes. If the associated TCA and TWA require 256 bytes and 500 bytes, respectively, the amount of dynamic storage in Subpool 1 available to transaction ABCD is: 10,000 - 3,756 = 6,244 bytes.

Several types of storage may be directly acquired from Subpool 1 by the application program through the use of the CICS Storage Control GETMAIN facility. These are:

1. Dynamic storage defined and controlled by the user
2. I/O area for the Temporary Storage facility
3. I/O area for the Transient Data facility

In addition, Subpool 1 storage is used by the CICS File Control program for file input/output areas and work areas, and by such CICS functions as ENQ/DEQ.

If any unconditional request for storage from Subpool 1 cannot be satisfied, the transaction is abnormally terminated since there is no possibility of storage being released in that subpool except by the transaction issuing the request. It is essential, therefore, that the size specified for Subpool 1 be large enough to accomodate the largest single transaction in the system.

Subpool 0

The size of Subpool 0 depends on four factors:

1. Size of partition
2. Size of Subpool 1
3. Size of CICS nucleus (static storage)
4. Number and size of resident application programs

A change in any of these factors may change the size of Subpool 0 storage, since it represents all the available storage within the partition after CICS static storage and Subpool 1 storage have been allocated and the specified application programs have been made resident.

Because of the nature of the data which resides in Subpool 0, it is very difficult to estimate the amount of storage required. However, the size of this subpool is not as critical to the execution of the user's application program since a "cushion" concept is used to help control storage allocation during periods of stress and system overload.

Unlike Subpool 1, an unconditional request for storage from Subpool 0 that cannot be satisfied causes the transaction to be temporarily suspended until some storage is released by another transaction. It should be apparent, however, that even though the lack of Subpool 0 storage does not abnormally terminate a transaction, it can seriously impair system performance. Therefore, it is desirable to have enough storage in Subpool 0 to adequately support the needs of the system at least 80% to 90% of the time.

The primary types of data located in Subpool 0 are Terminal I/O Areas, CICS control blocks, and the storage cushion.

Terminal Input/Output Areas

Terminal input/output areas are dynamically acquired by both CICS and the user's application. Input areas are acquired by CICS for polling and may also be acquired by the user's application when requesting terminal services. These input areas are 13 bytes in length plus the length of the area specified for the Terminal Control Table line entry (TCTLE). Output areas are acquired by the user's application when requesting terminal services and are 13 bytes in length plus the length specified by the user.

The number of terminal input/output areas simultaneously required in main storage is hard to ascertain, but generally there is a minimum of one input area per defined line and there may be as many as several per terminal, although this is an extreme situation. Input areas may also serve as output areas, so that only one area is used per terminal.

CICS Control Blocks

Subpool 0 is used to support the following CICS control blocks:

1. One Task Extension Area (TXA) per active transaction - 40 bytes.
2. One Queue Element Area (QEA) for each unique resource for which an enqueue has been requested - 16 bytes plus length of the symbolic resource name on which the enqueue is requested.

If the exclusive control feature of File Management is included in the system, each request for a data set update generates an enqueue request on a symbolic resource name consisting of the data set name concatenated with the logical record identification (physical record for DAM data sets).

The number of TXA's and QEA's simultaneously in Subpool 0 will depend on the peak activity expected.

Storage Cushion

In the CICS/DOS-ENTRY system, the storage cushion exists only in Subpool 0. This portion of main storage (size defined in the System Initialization Table) is acquired by CICS during system initialization and is used only when all other Subpool 0 storage has been exhausted.

The following are considerations in determining an adequate storage cushion size:

1. The cushion should be large enough to satisfy the largest request that will occur for Subpool 0 storage. In all probability, this request will be for a terminal I/O area.
2. The size of the cushion should not be so large that its acquisition reduces the size of the "normal" dynamic area below the level sufficient to meet the user's performance objectives.
3. The size of the cushion should not be so small that the release of the cushion is not sufficient to meet the needs of the system when it is under stress.

DYNAMIC STORAGE - CICS/DOS-STANDARD AND CICS/OS-STANDARD V2 SYSTEMS

Dynamic storage in the CICS/DOS-STANDARD and CICS/OS-STANDARD systems consists of essentially the same area as for the CICS/DOS-ENTRY system with the exception that resident application programs are part of dynamic storage rather than static storage. The storage organization for the CICS/DOS-STANDARD and CICS/OS-STANDARD systems is illustrated in Figure 17.

All characteristics applicable to Subpool 0 in the CICS/DOS-ENTRY system apply also to dynamic storage in the CICS DOS-STANDARD and CICS/OS-STANDARD systems except that Task Extension Areas (TXA's) are replaced with Task Control Areas (TCA's). In addition, all transaction-oriented data isolated in Subpool 1 of the CICS/DOS-ENTRY system is located in the dynamic storage area of the CICS/DOS-STANDARD and CICS/OS-STANDARD systems. This means that all data areas dynamically allocated to active transactions are in main storage for the duration of the transactions, rather than being rolled out to auxiliary storage as in the CICS/DOS-ENTRY system.

In the CICS/DOS-STANDARD and CICS/OS-STANDARD systems, nonresident application programs remain in main storage as long as there is enough dynamic storage to satisfy processing needs. Only when the system is under stress are non-active programs deleted.

If TCAM process queues are used in the CICS/OS-STANDARD systems, the amount of dynamic main storage acquired for terminal input/output areas is equal to the length specified in the input Terminal Control Table line entry.

From Figure 17 and the preceding discussion, it should be apparent that dynamic storage in the CICS/DOS-STANDARD and CICS/OS-STANDARD systems is significantly larger than in the CICS/DOS-ENTRY system.

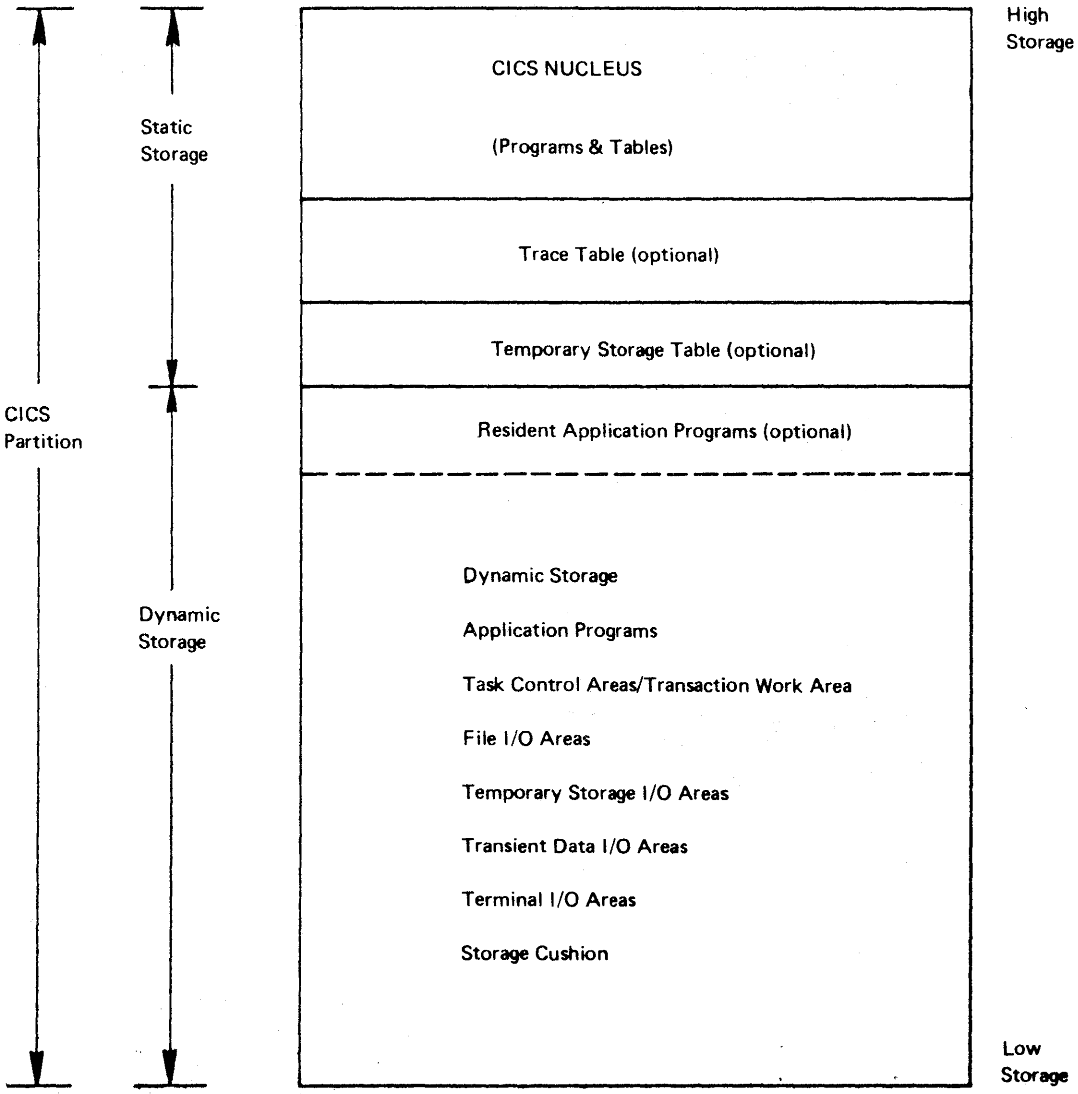


Figure 17. Storage organization for the CICS/DOS-STANDARD system and the CICS/OS-STANDARD system

STORAGE ESTIMATES WORKSHEET

STATIC STORAGE

PROGRAMS			TABLES		
Task Control	_____	(note 1)	CSA	_____	(note 1)
Storage Control	_____	(note 1)	PPT	_____	(note 1)
Program Control	_____	(note 1)	PCT	_____	(note 1)
Terminal Control	_____	(note 1)	TCT	_____	(note 1)
Interval Control	_____		FCT	_____	
Program Interrupt Control	_____		DCT	_____	
File Control	_____		TST	_____	
Transient Data Control	_____		TRT	_____	
Temporary Storage Control	_____		ROT	_____	(note 2)
Trace Control	_____				
Dump Control	_____				
Subtotal	_____	+	Subtotal	_____	= _____

ACCESS METHODS (note 3)			RESIDENT APPLICATION PROGRAMS		
System DAMOD	1500	(note 1)	(note 4)		
ISAM	_____				
BTAM	_____				
DL/I	_____				
TCAM	_____	(note 5)			
Subtotal	_____	+	Subtotal	_____	= _____

DYNAMIC STORAGE

Various factors enter into the determination of an adequate amount of dynamic storage for CICS. In attempting to estimate the dynamic storage requirements, the user must familiarize himself with the storage characteristics of the particular system he will be using as discussed in the sections on dynamic storage in this publication.

STATIC STORAGE ____ + DYNAMIC STORAGE ____ = TOTAL PARTITION SIZE ____

Note 1: Required in all systems.

Note 2: The Rollout Table is required and is applicable only in the CICS/DOS-ENTRY system.

Note 3: Necessary Sequential Access Methods (SAM) are included in the estimates for CICS/DOS programs and tables. The SAM requirement for CICS/OS programs and tables must be determined from the publication OS/360 Storage Estimates (GC28-6551).

Note 4: Resident application programs are considered part of static storage in the CICS/DOS-ENTRY system. For other CICS systems, they are part of dynamic storage.

Note 5: TCAM-dependent routines loaded into the CICS/OS partition/region require approximately 4.5K bytes. This estimate does not include the main storage required for the user-written TCAM Message Control program (which resides in another partition/region).

3270 NATIVE MODE SUPPORT

The 3270 Information Display System is a highly flexible device, both in its available hardware configurations and features and in its advanced programming and operator capabilities. The CICS 3270 Native Mode support is a basic read/write support that allows the user to take direct advantage of the functional capabilities of the 3270 system. The user, through operands of the DFHTC macro instruction, is able to request the following services:

1. Read operator input (operator-initiated transmission)
2. Read full buffer (program-initiated transmission)
3. Write
4. Erase/write
5. Erase all unprotected fields
6. Copy buffer (applicable cnly to remote 3271 Control Units)

The user is also able to control the following functions through the data stream and Write Control Character information supplied with write operations:

1. Screen format
2. Light pen detection
3. Cursor postion
4. Field display intensity
5. Print line format
6. Keyboard restore
7. Audible alarm
8. Modified data tag reset
9. Data protection
10. Start printing

The user has complete responsibility for the construction of valid 3270 output data streams and for the interpretation of 3270 input data streams. However, the 3270 Native Mode Support affords him a data stream transparency between local and remote configurations. The Terminal Control program automatically handles the input/output data transmission block length requirements of the remote 3270 systems, and also handles any data translation necessary before and after data tranmission. In addition, the Terminal Control program maintains certain control and statistical information concerning each terminal.

2260 COMPATIBILITY FOR 3270

2260-compatibility support for the 3270 Information Display System allows the user to run his currently operational 2260-based transactions from a 3270.

During CICS system generation, the user must request 2260 compatibility to be included, thereby generating the necessary code to provide conversion of 2260 data streams from user-written application programs to the appropritate 3270 data stream format. When the 3270 operates with a "compatibility" transaction, incoming data from the 3270 is converted and presented to the user-written application program in 2260 format. In most cases, no changes are required to the user-written program.

Because 2260 compatibility is specified by transaction as well as by terminal, non-2260-based transactions have full access to all facilities of the 3270. Only when a 2260-compatibile transaction is being run on a 2260-compatible 3270 does CICS perform the correct editing of the input and output data streams. If the transaction is not specified as 2260 compatible, or if the terminal is not specified as supporting 2260 compatibility, no editing occurs for the data streams. In that case, if the data streams are not valid 3270 data streams, the results are unpredictable.

Two modes of compatibility operation are provided: FORMAT and FULLBUF.

FORMAT mode takes full advantage of the 3270 formatting and data compression facilities, and is the preferred method of 2260 compatibility, particularly for the operation of remote 3270's. However, some 2260 facilities cannot be supported under FORMAT mode. For information concerning which facilities are available, see the discussion under the topic "Screen Techniques".

FULLBUF mode does not use the 3270 data compression facilities and must therefore be used when all lines of input data are desired. However, for each operator interaction involving a data entry key (ENTER, PF1 - PF12), the number of characters transferred is approximately equal to the total number of characters on the simulated 2260 screen. The exact number of characters transferred varies depending upon whether the 3270 is local or remote and which model of the 2260 is being mapped onto which model of the 3270.

For local 3270 operation, the extra data transfer of FULLBUF mode should be transparent to the user with regard to response time. For remote 3270 operation, the response time is a complex function of the present method of 2260 operation and the line speeds used for the 2260 and 3270; however, the increase in the response time (on a transaction basis) should be less than 20% at the same line speed.

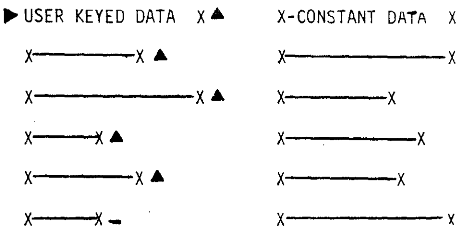
Individual transactions can be flagged for FORMAT or FULLBUF 2260 compatibility through the CICS DFHPCT TYPE=ENTRY macro instruction. The mode of compatibility chosen depends on the 2260 functions required for the application programs that run under this particular transaction code.

For planning purposes, it is estimated that approximately 3500 bytes of static storage are required for 2260/3270 compatibility support. In addition, approximately 30 bytes are required for each terminal defined as 2260 compatible plus the storage required for the operating system access method. 2260 I/O buffer space in dynamic main storage is estimated to increase by approximately a factor of two, depending on the application program design and the use of buffer space. (The variation is in the range from a few extra bytes per I/O buffer to a total buffer space increase by a factor of four.)

SCREEN TECHNIQUES

Various techniques have been used for entering data using a 2260 screen and keyboard. CICS 2260-compatibility support for the 3270 allows the use of four basic techniques:

1. Formatted 2260 screen technique; for example:



With this technique, the constant data is optional and is sent to the user at the start of the transaction. Either FORMAT or FULLBUF mode may be specified, depending upon the user's formatting requirements.

2. 2260 tab feature technique; for example:

▶ NAME:
INITIAL:
STREET:
NUMBER:
CITY:
STATE:

For CICS 2260 compatibility operation, the colon-tab character combination is replaced by 3270 "unprotected attribute" characters. FORMAT mode may be specified if data is always keyed into every field. FULLBUF mode must be specified if any field is to be left blank. (Unlike the 2260, the 3270 does not transmit blank characters unless FULLBUF is specified.) If FULLBUF is not specified, any heading following the blank field is not transmitted to the application program.

```
▶ NAME:    JONES          INITIAL:    J▲
  STREET:                 NUMBER:   1515▲
  CITY:  NEW YORK ▬       STATE:
```

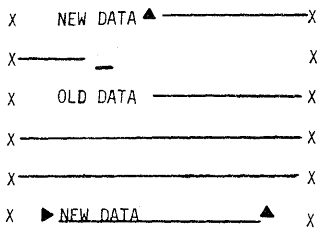
FORMAT data stream:

```
NAME:JONES        INITIAL:J(NL)         :1515(NL)       :NEW YORK
```

FULLBUF data stream:

```
NAME:JONES    INITIAL:J(NL)            STREET:       NUMBER:1515(NL)   CITY:NEW YORK
```

3. "Endless screen" technique; for example:



With this technique, the 2260 screen is treated as unformatted. The operator keys off the screen, and, wrapping around to the start of the screen, overkeys any old data still there. The 2260 transmits a data stream delimited by SMI (start of message) and EOM (end of message) characters, irrespective of any screen wraparound.

Either FORMAT or FULLBUF can be specified. In either case, CICS ensures that the data stream is correctly ordered before sending it to the 2260-based transaction.

With this technique, there is a difference in operation between FORMAT and FULLBUF modes only in the case of a 480-character 2260 mapped onto a 480-character 3270. Use of FORMAT mode causes the loss of the last character of every 2260 output line. Use of FULLBUF eliminates the data loss but at the expense of transferring a full 480 characters for each ENTER.

It is the responsibility of the user to determine whether his 2260 transaction can tolerate the loss of the last character of each output line. CICS appends a blank character to the end of each 2260 input line, except where the line is terminated prematurely by a new-line (NL) symbol.

4. "Change and enter" technique; for example:

```
►JONES        J  H
 1515         OLD ROAD EAST              ORIGINAL
 NEW YORK                                RECORD

 N.Y.   10010 _

►JONES        J  H
 1515         OLD ROAD WEST              UPDATED
 NEW YORK                                RECORD
 N. Y.  10010 _
```

The FCRMAT data stream looks like this:

```
JONES        J  H      1515    OLD ROAD WEST
```

The FULLBUF data stream looks like this:

```
JONES        J  H      1515    OLD ROAD WEST
NEW YORK               N.Y.  10010
```

The 2260 transaction sends an existing record to the screen. After making any necessary corrections to data fields, the operator depresses the ENTER key; the 2260 transaction receives the entire record in its updated version.

With this technique, FULLBUF must be specified for this transaction to ensure that the transaction receives a 2260-compatible data stream.

INITIATING TRANSACTIONS

The user can initiate 2260 compatibility transactions or 3270 native mode transactions, with or without prior clearing of the 3270 screen, by entering the appropriate four-character CICS transaction code. Any Start of Message Indicator (SMI) appearing before the transaction code is recognized by CICS; the succeeding four characters are interpreted as a CICS transaction code. CICS then initiates the specified transaction; in the case of a 2260 compatibility transaction, the 3270 screen is formatted automatically.

START OF MESSAGE INDICATOR (SMI)

For the 2260, X'4A' is displayed as the SMI (▶) character. If the SMI character is contained in an output data stream, it is displayed on the 3270 screen as follows:

Domestic	¢
U.K.	$
French	¢ or ç
German	ö

The cursor is positioned immediately following the SMI character. At the user's discretion, a different character may be specified during CICS system generation to represent the SMI. Whatever character is chosen, it remains the same for all transactions.

If an SMI character is not placed on the screen by the user's 2260 data stream, the cursor is positioned after the last character transmitted; the operator must then key an SMI character somewhere on the screen. Failure to do so results in no data being transmitted to the application program.

After the data has been read in, CICS 2260 compatibility transmits a single blank character to overwrite the SMI character.

NEW LINE SYMBOL (NL)

For CICS 2260 compatibility, the new line (NL) function is replaced by a field mark character followed by the NL key. Any incoming field mark characters are treated as 2260 NL characters, and the remainder of the line is discarded.

Including the NL symbol in the 2260 output data stream causes the substitution of a 3270 field mark character; the output continues at the beginning of the next line. (The 3270 field mark character displays as a semicolon.)

LINE ADDRESSING

For a 3270 operating under CICS compatibility, any requests for write at line address are honored. CICS 2260 compatibility converts the line address to the equivalent 3270 buffer address before transmitting the data stream to the screen.

2848 LOCK FEATURE

The optional lock feature available on the 2848 Control Unit Models 21 and 22 is supported by CICS 2260 compatibility. Any read/lock request is honored by CICS 2260 compatibility by leaving the 3270 keyboard inhibited. The keyboard is reset only if the transaction changes or if a keyboard RESET request is issued by the application program.

2260 MODEL-DEPENDENT DATA STREAM

Some users require the capability of building device-dependent data streams. CICS allows the user to build such data streams by providing the terminal type at TCTTETT in the Terminal Control Table (TCT) and the terminal model number at TCTTEMN in the TCT.

The TCTTETT and TCTTETM fields always contain the 2260 or 2265 terminal-type codes and user-assigned model number (as specified in the DFHTCT TYPE=TERMINAL macro instruction) whenever a transaction flagged for 2260 compatibility is dispatched. At all other times, TCTTETT and TCTTETM contain the codes for the 3270 terminal.

3270 BASIC MAPPING SUPPORT

CICS provides Basic Mapping Support (BMS) for use with the IBM 3270 Information Display System. Through BMS the CICS application programmer has access to input and output 3270 data streams without the need to include any 3270 device-dependent code in the CICS application program.

Maps are assembled offline through use of CICS macro instructions. The user defines and names fields and groups of fields which may be written to and received from the 3270. The assembled maps contain all the 3270 device-dependent control characters necessary for the 3270 data stream.

Associated with each map is a table of field names which is copied into each application program that uses the map. Data is passed to and from the application program under these field names. The application program is written to manipulate the data under the various field names so that alteration of a map format on the 3270 does not necessarily alter the program logic. New fields may be added to a map format without making it necessary to reprogram existing applications.

Output data may be supplied from the application program by placing the data in the table under the appropriate field name. As an alternative, output maps may contain field default data which is sent when data supplied by an application program is not present. This facility permits the specification of titles, headers, etc. for output maps.

Optionally, the displaying of all default data may be suppressed by the application program for any output map. Each time a map is used, the application program may temporarily modify the attributes of any named field in the output map. Output map fields with no field name may contain default data, but the application program can never replace the default data or modify the attributes of these unnamed fields.

For input maps, the user assembles a map which defines the fields that are potentially receivable from the 3270. Any data received for a particular field is moved across under the correct field name in the symbolic storage definition for this map. Any pen-detectable fields defined within an input map are flagged as detected if present in the 3270 input stream. An input map may specify a subset of the fields potentially receivable; any received fields not represented in the current input map are discarded. This permits the number of keyable and selectable fields from a map to be changed without making it necessary to reprogram applications that currently receive data from that map.

Maps are stored in the CICS program load library, or, in the case of Assembler language, the maps may be coded in the application program. Whenever a map stored in the program load library is referenced by BMS, a copy is automatically retrieved by CICS without any application program action. Multiple users of a map contained in the program load library share a single copy in main storage.

BMS permits any valid combination of field attributes to be specified by the user when generating maps. Inclusion of BMS in CICS is a system generation option and its inclusion does not prevent the application program from access to the 3270 in native mode. Intermixing the use of BMS and native mode support for a terminal from the same application program may yield unpredictable results. When using mixed mode support, it is the users' responsibility to ensure the correct construction and interpretation of the native mode 3270 data streams.

BMS permits the application program to pass a 3270 native mode data stream (already read in) and map this data stream into a given input map. This facility allows data entered with the initial read of a transaction to be successfully mapped via BMS.

Estimates of the main storage required by Basic Mapping Support (BMS) are as follows:

1. Mapping transaction module DFHBMSMM: Approximately 1.5K bytes.

2. User-supplied TIOA for data transmission to output map:

   Size = S + 2N

   where S is the sum of the lengths of all named fields (or groups of fields) and N is the total number of fields.

3. TIOA acquired by BMS for output map:

   Size = S = 5N + 4

   where S is the sum of the lengths of all named fields (or groups of fields) and N is the total number of fields. The calculated size is the absolute maximum for a set of non-contiguous fields.

4. TIOA obtained by BMS to pass back to user for input map:

   Size = S + 2N

   where S is the sum of the lengths of all named fields (or groups of fields) and N is the total number of fields.

5. TIOA obtained by BMS to read in 3270 data:

   Size = S + 5N + E

   where S is the sum of the lengths of all named fields (or groups of fields), N is the total number of fields, and E is the number of characters keyed in beyond the maximum field length specified by the user in the DFHMDF macro instruction. Note that the TIOA size specified by the user in the TCTTE for 3270's must be large enough to contain the largest data stream required for a mapping operation.

6. Storage required by maps:

   Output map size = 14 + 5N + S
   Input map size = 12 + 4N

   where N is the total number of fields (or groups of fields) and S is the sum of all initialized field lengths.

GLOSSARY

Device Dependent. The CICS program characteristics that are designed to function only for a specific device.

Dispatch. The transfer of control of the central processing unit (CPU) to a task.

Event. An occurrence of significance to a task; typically, the execution of an asynchronous operation, such as input/output.

Polling Interval. The period of time between one attempt to read data from a terminal or group of terminals and the next attempt to read data from the same terminal or terminals.

Serviceability. The characteristics of a data processing system that are designed to provide statistics and diagnostic aids for system repair and preventive maintenance.

Synchronize. To indicate that an event is coincident with another event.

Transaction. A logical system operation that may be composed of one or more inputs and outputs.

BIBLIOGRAPHY

CICS PUBLICATIONS

Application Programmer's Reference Manual	SH20-1047
System Programmer's Reference Manual	SH20-1043
Terminal Operator's Guide	SH20-1044
CICS/DOS Operations Guide	SH20-1034
CICS/OS Operations Guide	SH20-1048
Logic Manual for CICS/DOS-ENTRY	LY20-0712
Logic Manual for CICS/DOS-STANDARD	LY20-0713
Logic Manual for CICS/OS-STANDARD V2	LY20-0714

DOS/360 PUBLICATIONS

IBM System/360 Disk and Tape Operating Systems Concepts and Facilities	GC24-5030
IBM System/360 Disk Operating System Basic Telecommunications Access Method	GC30-5001
IBM System/360 Disk Operating System Supervision and Input/Output Macros	GC24-5037
IBM System/360 Disk Operating System System Generation and Maintenance	GC24-5033
IBM System/360 Disk and Tape Operating Systems Assembler Language	GC24-3414
IBM DOS American National Standard COBOL	GC28-6394
IBM System/360 Disk Operating System Full American National Standard COBOL Programmer's Guide	GC28-6398
IBM DOS American National Standard Subset COBOL	GC28-6403
IBM DOS Subset American National Standard COBOL Compiler and Library Programmer's Guide	SC28-6439
IBM DOS Full American National Standard COBOL Compiler and Library, Version 3, Programmer's Guide	SC28-6441
DOS PL/I Optimizing Compiler Programmer's Guide	SC33-0008

OS/360 PUBLICATIONS

IBM System/360 Operating System Introduction	GC28-6534
IBM System/360 Operating System Assembler Language	GC28-6514

IBM System/360 Operating System Graphic Programming Services for 2260 Display Station	GC27-6912
IBM System/360 Operating System, Basic Telecommunications Access Method, Version 2	GC30-2004
IBM System/360 Operating System Generation	GC28-6554
IBM System/360 Operating System, Multiprogramming with a Fixed Number of Tasks - Version 2	GC27-6939
IBM System/360 Operating System Concepts and Facilities	GC28-6535
IBM System/360 Operating System Supervisor and Data Management Services	GC28-6646
IBM System/360 Operating System, Full American National Standard COBOL	GC28-6396
IBM System/360 Operating System, Full American National Standard COBOL Programmer's Guide	GC28-6399
IBM OS Full American National Standard COBOL Compiler and Library, Version 3, Programmer's Guide	SC28-6437
IBM OS Full American National Standard COBOL Compiler and Library, Version 4, Programmer's Guide	SC28-6456
IBM System/360 Operating System PL/I F Programmer's Guide	GC28-6594
PL/I Optimizing Compiler Programmer's Guide	SC33-0006
IBM System/360 Operating System Storage Estimates	GC28-6551
IBM System/360 Supervisor and Data Management Macro Instructions	GC28-6647
IBM System/360 Operating System TCAM Programmer's Guide and Reference Manual	GC30-2024

IMS PUBLICATIONS

General Information Manual	GH20-0765
System/Application Design Guide	SH20-0910
System Programming Reference Manual	SH20-0911
Application Programming Reference Manual	SH20-0912
Utilities Reference Manual	SH20-0915

READER'S COMMENT FORM

Customer Information
Control System (CICS)
General Information Manual

GH20-1028-3

Please comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers). All comments and suggestions become the property of IBM. If you wish a reply, be sure to include your name and address.

COMMENTS

fold

fold

fold

fold

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
FOLD ON TWO LINES, STAPLE AND MAIL.

Your comments, please . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold Fold

First Class
Permit 40
Armonk
New York

Business Reply Mail

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

Postage will be paid by:

International Business Machines Corporation
Department 813
1133 Westchester Avenue
White Plains, New York 10604

Fold Fold

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

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

CICS GIM Printed in U.S.A. GH20-1028-3

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

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