IBM

CICS

CICS

CICS

CICS

General Information

IBM

CICS
CICS
CICS
CICS

Customer
Information
Control System

Program Product
Version 1.6 and 1.7

Program Numbers
5740-XX1 (CICS/OS/VS)
5746-XX3 (CICS/DOS/VS)

General Information

GC33-0155-2

Third Edition (July 1985)

This edition applies to Version 1 Release 7 (Version 1.7) of the IBM program product Customer Information Control System/Operating System/Virtual Storage (CICS/OS/VS), program number 5740-XX1, and to Version 1 Release 6 (Version 1.6) of the IBM program product Customer Information Control System/Disk Operating System/Virtual Storage (CICS/DOS/VS), program number 5746-XX3, and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters.

The CICS/OS/VS user should note that this edition replaces GC33-0155-1, which applied to CICS/DOS/VS Version 1 Release 6 and to CICS/OS/VS Version 1 Release 6.

Users of CICS/OS/VS and CICS/DOS/VS Version 1 Release 6 can still obtain the second edition of this book by using the following order number: GT33-0155.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the addresses given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed either to:

International Business Machines Corporation, Department 6R1H,
180 Kost Road, Mechanicsburg, PA 17055

or to:

IBM United Kingdom Laboratories Limited, Information Development,
Mail Point 095, Hursley Park, Winchester, Hampshire, England, SO21 2JN .

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

What this Book is About

This book describes CICS, the Customer Information Control System. CICS is an IBM program product used for data-base/data-communication. This edition of the book is about both CICS/OS/VS Version 1 Release 7, first available in 1985, and CICS/DOS/VS Version 1 Release 6, first available in 1982.

Who this Book is For

This book is for:

Data processing managers
System designers
Application designers

What you Need To Know to Understand This Book

This book assumes some experience of computer applications, but we don't assume you know anything about CICS.

How to Use this Book

We would like you to read this book sequentially. You can use the appendixes for reference.

Notes on Terminology

"CICS" is used throughout this book to mean both CICS/OS/VS and CICS/DOS/VS generally (except when we want to point out specific differences between them).

Book Structure

Bibliography

CICS/OS/VS Version 1 Release 7 Library

General

- General Information GC33-0155
- Library Guide GC33-0356
- Release Guide GC33-0132
- Master Index SC33-0227
- Messages and Codes SC33-0226

Planning

- Facilities and Planning Guide SC33-0202
- Intercommunication Facilities Guide SC33-0230
- Recovery and Restart Guide SC33-0231
- Performance Guide * SC33-0229
- Performance Data * SC33-0212
- 3270 Data Stream Device Guide SC33-0232
- 3650/3680 Guide SC33-0234
- 3767/3770/6670 Guide SC33-0235
- 3790/3730/8100 Guide SC33-0236
- 4700/3600/3630 Guide SC33-0233

Administration

- Installation and Operations Guide SC33-0071
- Resource Definition (Online) SC33-0186
- Resource Definition (Macro) SC33-0237
- Customization Guide SC33-0239
- CICS-Supplied Transactions SC33-0240

Programming

- Application Programming Primer SC33-0139
- Application Programmer's Reference (Command Level) SC33-0241
- Application Programmer's Reference Summary (Command Level) GX33-6047
- Application Programmer's Reference (Macro Level) SC33-0079

Service

- Problem Determination Guide SC33-0242
- Program Debugging Reference Summary SX33-6048
- Data Areas LY33-6035
- Diagnosis Reference LC33-0243

* To be published after general availability

CICS/DOS/VS Version 1 Release 6 Library

General

General Information	GC33-0155
Library Guide	GC33-0356
Release Guide	GC33-0130
Master Index	SC33-0095
Messages and Codes	SC33-0081

Planning

System/Application Design Guide	SC33-0068
Intercommunication Facilities Guide	SC33-0133
Recovery and Restart Guide	SC33-0135
Performance Guide	SC33-0134
Performance Data	SC33-0219
3270/8775 Guide	SC33-0096
3650/3680 Guide	SC33-0073
3767/3770/6670 Guide	SC33-0074
3790/3730/8100 Guide	SC33-0075
4700/3600/3630 Guide	SC33-0072

Administration

Installation and Operations Guide	SC33-0070
Resource Definition Guide	SC33-0149
Customization Guide	SC33-0131
Operator's Guide	SC33-0080

Programming

Application Programming Primer	SC33-0139
Application Programmer's Reference (Command Level)	SC33-0077
Application Programmer's Reference Summary (Command Level)	GX33-6012
Application Programmer's Reference (RPG II)	SC33-0085
Application Programmer's Reference (Macro Level)	SC33-0079

Service

Problem Determination Guide	SC33-0089
Program Debugging Reference Summary	SX33-6010
Data Areas	LY33-6033
Diagnosis Reference	LC33-0105

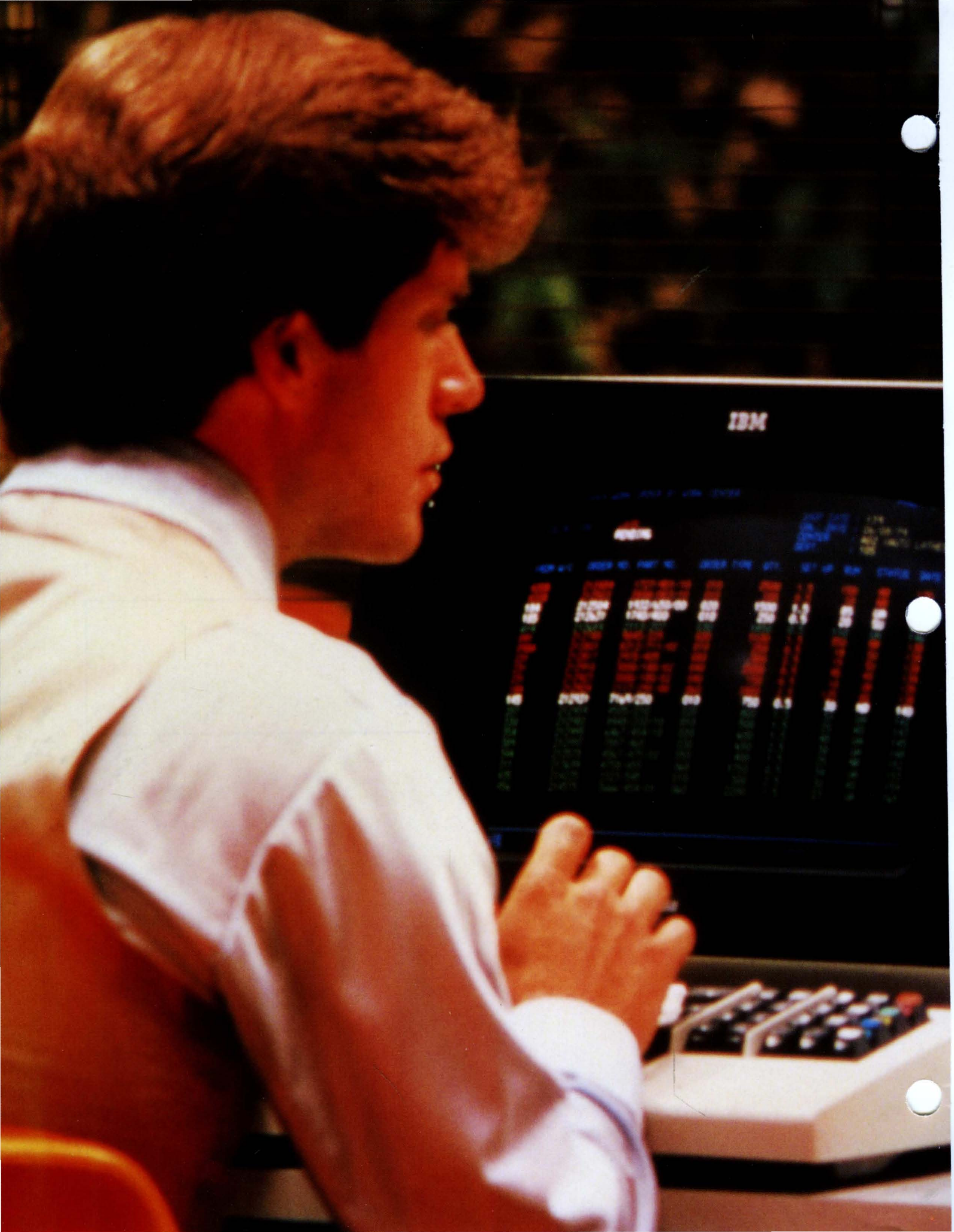
Contents

Figures

Part One

This Part of the book describes:

- Online systems
- The advantages of using CICS
- Typical CICS application programs


IBM

What is CICS?

Customer Information Control System (CICS) is a general-purpose mainframe-based data communication system — an online monitor, and some batch utilities — that can support a network of many hundreds of terminals. It's designed to support commercial "business" transactions, rather than scientific or CAD/CAM work.

You may find it helpful to think of CICS as an operating system within your own operating system (although this definition might offend purists). In these terms, CICS is a specialized operating system whose job is to provide an environment for the execution of your online application programs, including interfaces to files and data base products. See Figure 1.

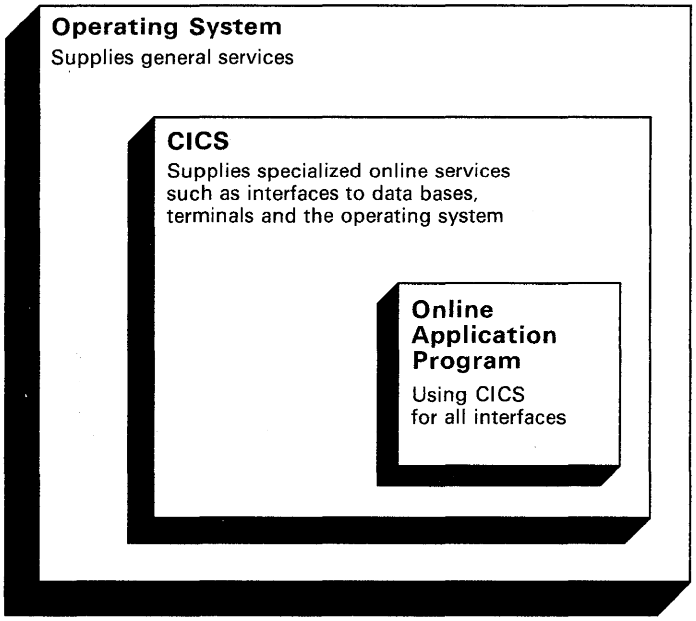

Operating System
Supplies general services
CICS
Supplies specialized online services such as interfaces to data bases, terminals and the operating system
Online Application Program
Using CICS for all interfaces

Figure 1. The CICS Environment

Why You May Need an Online System

In the 1970s (and even the early 1980s) most programs were batch programs. Typically, they would read a file, process individual data records, update a carried-forward version of the file, and produce some type of printed output. These files would usually go offline when the program had finished with them, and the file data thus became inaccessible for inquiry purposes. To make matters worse, the records in the files would only be as up-to-date as the most recent program run, and wouldn't reflect any intervening (non-computer) activity.

Nowadays, this often isn't good enough. Your users want immediate responses to their information processing needs. The overnight turnaround associated with traditional batch systems is no longer adequate: accurate, up-to-date information is needed within seconds. For this you need an online information processing system, using terminals that can give direct access to data held in either data sets or data bases. In other words, you need a **data-base/data-communication system**, or **DB/DC system**.

Your host operating system, of course, is still the final interface with the computer; CICS is "merely" another interface, this time with the operating system itself.

Operating systems are designed to make the best use of the computer's various resources. CICS helps out by separating a particular kind of application program (namely, online applications) from others in the system, and handling these programs itself.

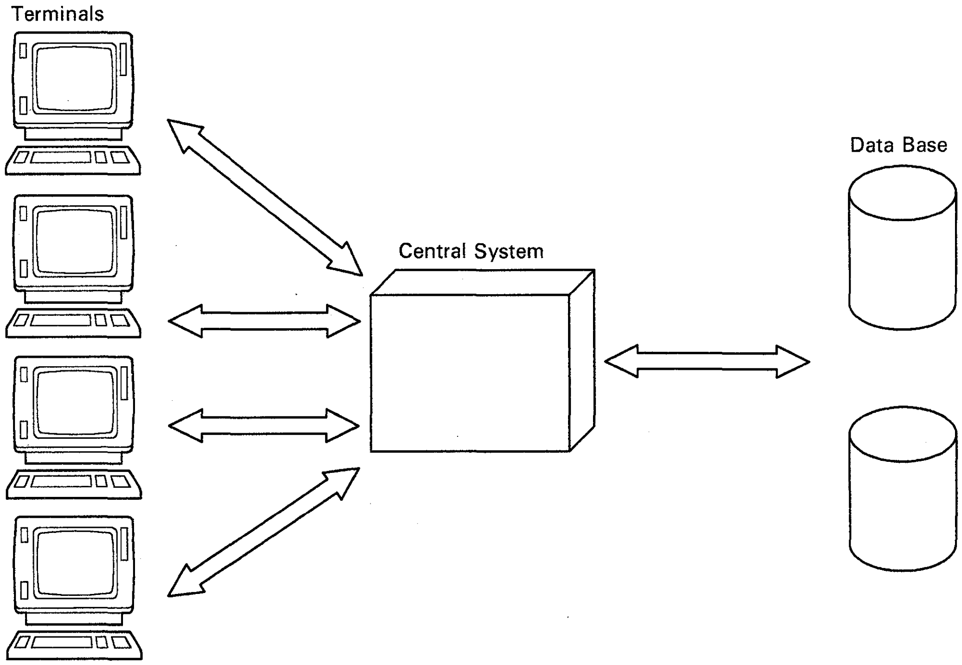


Figure 2. A DB/DC System

Developing a DB/DC system can be a major undertaking, particularly if you choose to write all your own control programs for handling terminals and files, and to provide your own job-scheduling mechanisms. However, CICS can make it very much easier by supplying all the basic components needed to handle your data communications. This allows your people to concentrate on developing application programs to meet your organization's business needs. They don't need to concern themselves with the details of data transmission, buffer handling, or the properties of individual terminal devices.

Why Have CICS?

The online end users within a network make all sorts of demands on many different sets of data. The things they want to do individually (for example, add a new employee to the payroll file) are usually simple and don't take very long. Often their jobs are interrelated and share the same programs and data. Furthermore, the response times that the users get should be as short as possible. For all these reasons, the users' work is done more efficiently within a single operating system job, rather than as separate jobs.

If all the work is to be handled within the same job, you need a controller to look after it, in much the same way that an operating system is needed within a computer to control the jobs. CICS carries out this controlling function within a DB/DC job.

CICS provides the communications control and service functions necessary for users to create their own, customized DB/DC system. This cuts down the amount of programming needed.

CICS is actually two things: the software that you install on your computer system, and the shelf full of books describing that software. It supports a very wide range of terminals: printers, monochrome and color displays, badge readers, banking devices, and so on. Even PCs can be attached in 3270 mode. And one CICS system can talk to others, both in the same host and at the other side of the globe. So you can base a world wide DC system around CICS. You don't have to, though. CICS is equally at home with small clusters of local devices.

CICS has gained wide customer acceptance over the years, one result of which is the large number of software products (both IBM and non-IBM) that will run with, or under, CICS. These range from application generators to performance monitoring tools, and from simple utilities to complete off-the-shelf applications that can go more or less straight into production on your system.

IBM tries to maintain a high degree of application compatibility across releases. We also do what we can to improve CICS usability, and installation, by listening carefully to suggestions and feedback from our users. CICS can meet the needs of practically any online application, since it can support a wide variety of terminals and subsystems.

We've left things as open as possible to allow our customers to produce the system **they** need. It's up to your system and application designers (which could mean you, of course) to choose what they want from the various CICS facilities, and to build whatever kind of user interface best suits the end users. So, although you still have to provide the application programs that the end users actually run, CICS makes them much easier to write. Your programs gain access to the CICS facilities they need by straightforward, high-level, commands. (The so-called "command-level interface.")

Couldn't I Do All This Myself?

Yes, of course, but why reinvent the wheel?

CICS is a large, mature piece of software that has evolved in parallel with the growth of online terminal networks and the movement toward distributed processing. It supports a wide range of hardware and software (for details, have a look at Appendixes A and B or at the *CICS/OS/VS Facilities and Planning Guide*). Indeed, many thousands of data-processing installations (both large and small) around the world have based their data communication systems on CICS.

What Does CICS Do?

CICS controls online DB/DC application programs. But what does this mean? In fact, it means that CICS does a lot of work on your behalf. CICS handles interactions between terminal users and your application programs. An interaction may consist of one or more requests from, and responses to, a terminal user in the course of a single job by that user.

CICS provides:

- The functions that application programs need for communication with remote and local terminals and subsystems
- Control of concurrently running programs serving many online users
- Access to data bases and files, in conjunction with the various IBM data base products and data access methods that are available
- Communication with other CICS systems and data base systems, both in the same computer and in connected computer systems.
- Many other facilities essential to the creation, operation, and maintenance of an online system.

Your host operating system remains in overall charge of the computer and manages resources in whatever way you set up. But the very versatility of a general-purpose operating system sometimes stops it giving online programs the sort of priority treatment they need. However, CICS can be given "privileged" treatment on behalf of all the online programs that run under it.

So, CICS takes on itself some of the aspects of an operating system. For example, CICS allows more than one of its programs to be in an active state at the same time. But CICS doesn't duplicate all of the services provided by the operating system. Whenever appropriate, CICS goes straight to the operating system to provide what its programs ask for.

Because CICS **is** a general-purpose product, the view your users get of it depends far more on the configuration of your system and its application programs than on any intrinsic features of CICS.

CICS Application Programs

Nowadays, terminal users connected to CICS remain largely unaware that CICS is controlling the scheduling of their applications along with that of other applications for other terminal users. They spend their time using the online application programs designed for their particular transactions. (This is as it should be, of course.)

CICS application programs typically:

- Serve many online users, apparently simultaneously
- Provide common access to the same data sets and data bases
- Give each end user a timely response to each interaction
- Involve telecommunications access to remote terminals.

How Does a CICS-Based Application Differ from a Batch Application?

Not **everything** is different, of course. But here are some basic differences for you to think about:

- In a batch program, all the required input/output and work areas are often defined within the program. In CICS, these areas are outside the program. They are allocated by CICS, as needed, from a dynamic storage area within the CICS partition or region. This lets CICS economize on main storage, and use the same copy of a program to do work for several users at once.
- A batch program reads its own input data, whereas CICS reads the data on behalf of the CICS application programs. A particular CICS application program need not even be loaded into the computer before its first input message arrives.
- A batch program issues its input/output instructions directly to the operating system. CICS application programs always issue such instructions to CICS, and CICS handles the interface to the operating system.
- Recovering when things go wrong is more complex with CICS applications — see "Data Integrity" on page 33.

Perhaps the most striking difference is how a small, simple application program can be loaded into the computer and can promptly be used by hundreds of people throughout a terminal network, all at the same time. And yet these online application programs are no more difficult to write and get working than the batch programs you've been used to in the past.

Application programs are stored in a library on a direct access storage device (DASD) attached to the processor. They can be loaded when the system is started, or simply loaded as required. If a program is in storage and isn't being used, CICS can release the space for other purposes. When the program is next needed, CICS loads a fresh copy of it from the library.

Two Vital Terms

Two important words in the CICS vocabulary are "transaction" and "task." You'll often see these in the CICS books, so it's good to know what they mean right from the start.

A **transaction** is a piece of processing initiated by a single request, usually from an end user at a terminal. A single transaction will consist of one or more application programs that, when run, will carry out the processing needed.

Normally, end users wishing to begin an online session will first identify themselves to CICS by signing on. Signing on to CICS gives users the authority to invoke certain transactions. Once signed on, they invoke the particular application (transaction) they intend to use. They can do so by typing the **transaction identification code** at the start of their initial request. But, if your designers decide otherwise, it's just as easy to set up a particular program function (PF) key to invoke a transaction with a single keystroke or, indeed, for a given terminal always to invoke a particular transaction.

CICS looks up the transaction identifier in one of its internal tables, the Program Control Table, where it finds out which program to invoke first to do the requested transaction.

And **task**?

The CICS activity to execute a program for a user is called a **task**. A task is, in the simple case, the execution of a transaction.

A task can read from and write to the terminal, read and write files, start other tasks, and do many other things. All these services are controlled by and requested through CICS commands in your application programs. CICS manages many tasks concurrently. The number of tasks that can actually be executing at any one instant depends on the characteristics of the processor. However, when a task requests a service which involves a wait, such as file input/output, CICS uses the wait time of the first task to execute a second. To the users, it looks as if many tasks are being executed at the same time.

Typical Applications

Under the next few headings, we'll see how CICS can meet the needs of a typical application — the online requirements of a department store. But, first of all, what **are** these requirements? Have a look at the boxes in Figure 3. (Incidentally, the *CICS/VS Application Programming Primer* contains a COBOL implementation of this application.)

Nearly all DB/DC applications fall into one or other of the following categories:

- Online inquiry
- Online inquiry and update

- Online data entry
- Online message switching.

Application programs running under CICS can handle all such types of application at the same time. Different online applications can access the same data set or data base, and can also access data sets or data bases on other systems. (See "Intercommunication" on page 27.)

The Department Store's Current Situation

A department store with credit customers keeps a master file of its customers' accounts. Each customer record holds the customer's name, address, telephone number, charge limit, current balance, account activity, payment history, and so on. Currently, a set of batch processing programs updates this file (and related ones) with the necessary charge and payment information.

Online Access to Information

The store wants online access to a customer's record, to have absolutely current information. In addition, the Accounting Department wants to update these customer records online, for convenience and currency. This means adding new records, deleting records, and changing addresses and other information not related to billing.

Each customer has a unique account number, which is the key to the existing master file. The users in the Accounting Department access records by this number, because it is always available when they are processing work or researching questions.

The Customer Service Department wants to be able to access the file by customer name. If customers want to charge items but don't have their charge cards with them, a clerk will call Customer Service, verify the existence and payment status of the account, and get the account number for the charge slip.

Figure 3. A Department Store's Online Requirements

Online Inquiry

Online inquiry provides you with ready access to information in centralized files (the data base). You enter an inquiry and the application replies with a message containing data retrieved from the data base.

With most applications, some users will need only to retrieve information, without needing to change it. Online inquiry is for them. See Figure 4.

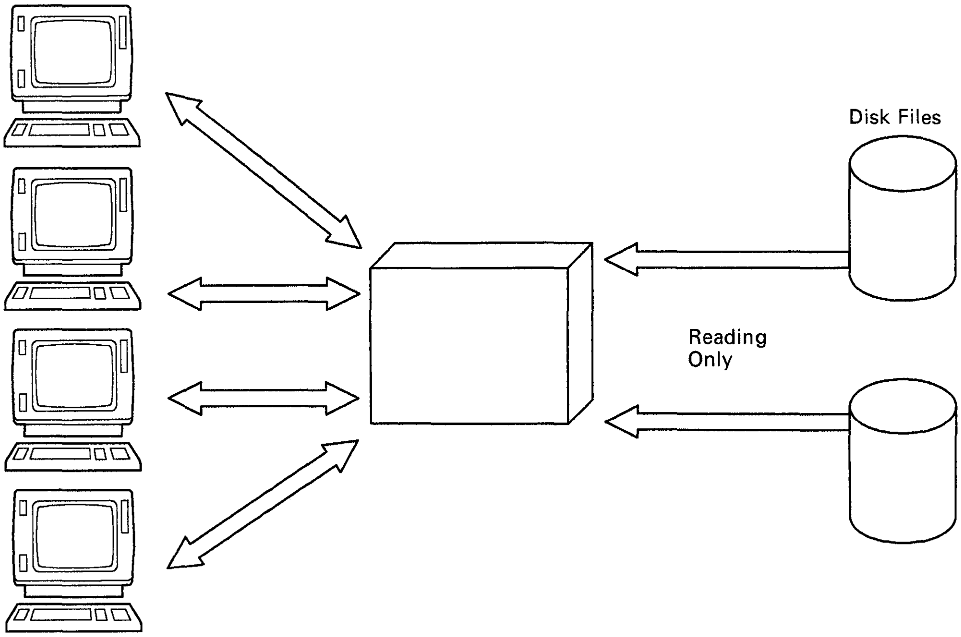


Figure 4. Online Inquiry

In a department store, for example, all the sales clerks need online inquiry when making customer credit account inquiries.

Online inquiry is the simplest type of application, mainly because data files are merely read (or perhaps browsed), not altered. This helps keep response times short.

Online Inquiry and Update

Online inquiry and update not only lets you read the data base; it also enables you to alter, delete, and store data in it.

The department store needs it; many of the clerks in the Accounting Department are constantly updating customer records online. Figure 5 shows inquiry and update.

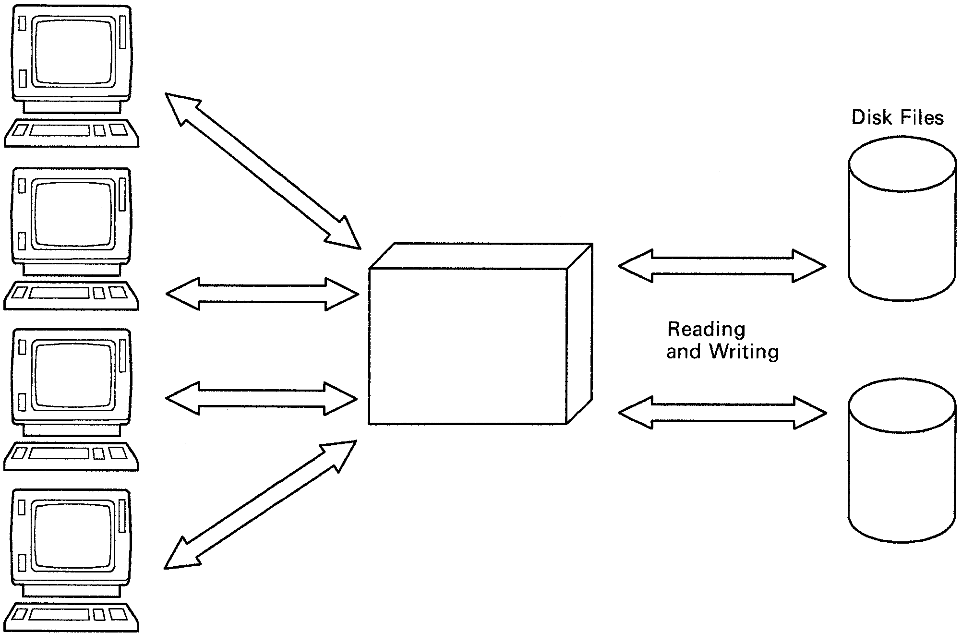


Figure 5. Online Inquiry and Update

Although this type of application also requires short response times (ideally), it involves rather more than a simple inquiry. There's a need to protect the central files, with safeguards against system failure during the update process and against the simultaneous updating of the same data item by more than one user. (All these requirements are dealt with in the *CICS/OS/VS Facilities and Planning Guide.*)

Online Data Entry

Online data entry allows data to be entered rapidly, with minimum interruption. It's used to create files at the central location for subsequent processing or access by other system users. It often replaces a batch system in which data entered offline is read into a data set in a single batch operation. In our example, details of new customers can be placed on the system for immediate use by the sales clerks.

How requests for data are made, and how the operator has to enter data, depends entirely on the application program. It may allow the operator to enter data quickly, according to some preset format, and with little or no prompting. Alternatively, it may provide quite a lot of help, making it easier (but perhaps slower) to enter correct data.

Typically, the application checks each line of data, and responds to the user only to point out invalid entries. It requires very short response times so that the user can continue data entry without delay, or correct invalid entries with minimal disruption. See Figure 6.

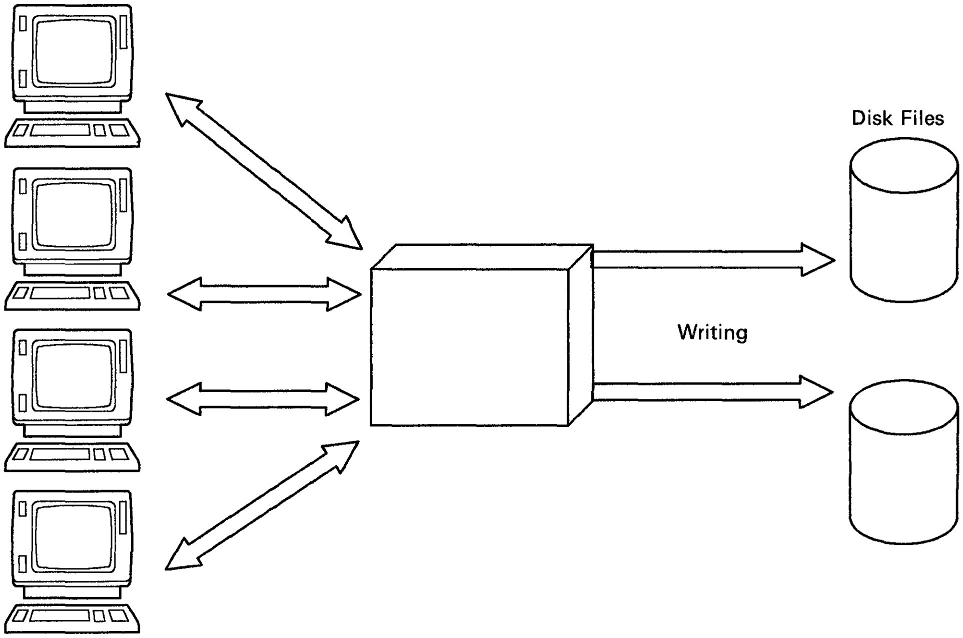


Figure 6. Online Data Entry

Whatever the degree of prompting, the application program can validate the entered data to the extent required. Once stored, the data can be made immediately available for access by other authorized terminals and users.

Online Message Switching

Online message switching handles the transfer of messages between terminals in the network. CICS can store messages if the recipient isn't available. Messages can also be held until a particular time is reached. There's no data processing or access to central files. Figure 7 shows this.

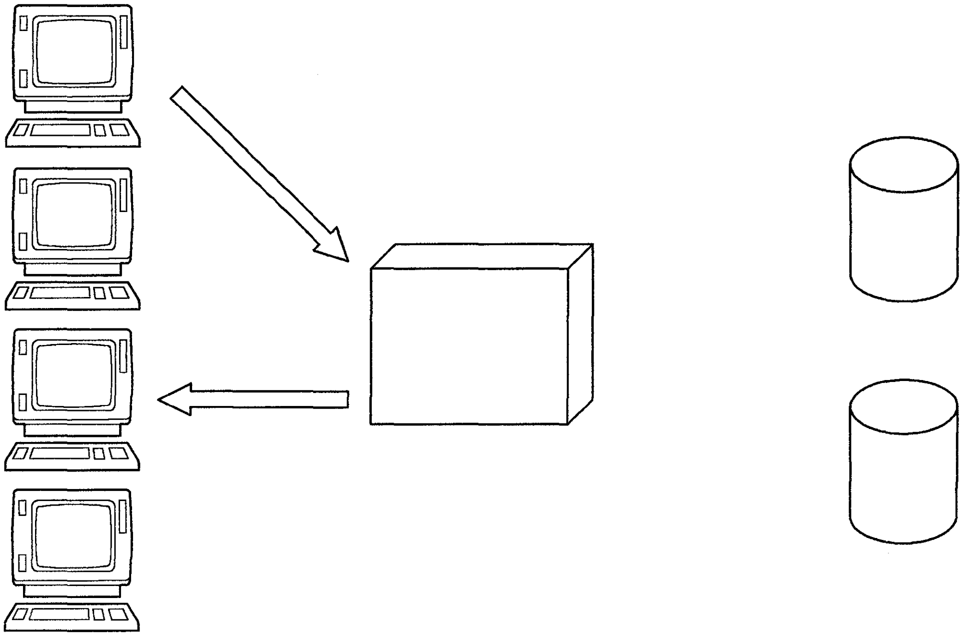


Figure 7. Online Message Switching

How Does CICS Help You Set Up an Online System?

After your system has been designed, the programming effort to turn the specification into a working reality is normally divided between two groups: the people who install and maintain the system, and those who write the application programs it will use. (Of course, all this work may be done by one person.) CICS offers a variety of helpful features for both groups.

Application Programming Aids

- A **choice of programming language**. Your application programs can be written in COBOL, PL/I, RPGII (under VSE only), or assembler language. (More details are in Part Three.)

- A **command-level programming interface** with CICS. Application programmers need know little about how CICS works: they request data or communication with terminals by issuing CICS commands similar in appearance to those of the programming language they are using. (For example, EXEC CICS READ..., EXEC CICS HANDLE CONDITION...) An offline **command language translator** takes their application source code, and translates the CICS commands into the appropriate language statements. It also provides useful diagnostics.

- **Basic Mapping Support (BMS)** which provides both device independence and format independence for 3270-system display and printer terminals. Device independence means the application programmer needn't know the control characteristics of the terminal. Format independence simplifies the positioning of data on the terminal, and allows data displays to be adapted for different terminals without changing the program.

- **Device and data independence**. Because CICS handles all communications and data accesses, application programmers do **not** have to handle the specific data formats for individual terminals or data storage devices.

- An **execution diagnostic facility (EDF)**. This CICS-supplied transaction enables programmers to test command-level application programs interactively. (Very useful, for example, when a programmer wants to exercise an error path by simulating a bad return code from some event.)

- A **command interpreter** facility. This CICS-supplied transaction enables programmers to enter and check the syntax and execution of individual commands interactively. (Very useful if a programmer forgets precisely which operands apply to a particular command.)

- For those who still need it, there's also a **macro-level programming interface** — see "CICS Macro-Level Programming" on page 58.

System Programming Aids

- **Pregenerated systems** to speed up and simplify the generation of the CICS system.

- Precompiled **sample programs** to help verify that the system is correctly installed.

- **Control tables** to make the system flexible. These allow you to define the terminals, files, and applications running under the CICS system. You can change the tables when required **without** changing either CICS or the application programs. And you can change some of them online, while CICS is up and running.

- **Journaling facilities** that allow you to create sequential journals for recovery, audit trail, and other purposes -- see the next item.

- **Recovery functions** to assist recovery from a variety of possible errors. Your system programmer can tailor the recovery facilities to suit your particular installation's needs.

- **User exits** that allow your system programmer to customize your CICS system by putting in routines that can take control under specified circumstances.

- **Monitoring facilities** that allow CICS to measure its own performance and that provide statistics to help tune a CICS system.

- **Security facilities** to help prevent unauthorized access to your information. Your system programmer can also tailor the security facilities, if necessary.

- **System services** required by online applications. These include queuing and temporary data storage, performance monitoring, activity statistics, trace and dump utilities.

- **Modularity** to allow you to include only those functions needed to support a particular set of applications.

Remember, too, (as we said earlier) that IBM always seeks to maintain application program compatibility (at both source and object level) from one release of CICS to the next. This means you can grow your CICS system, adding functions and facilities as your information processing needs grow -- with minimal impact on your existing investment in application programs.

What's next?

In the next part of the book, we move on from our overview of CICS, and tell you more about what goes on *inside* a CICS system.

Part Two

This Part of the book describes:

- The flow of control during a CICS transaction
- CICS data communication functions
- CICS data handling functions
- Application program services
- System services

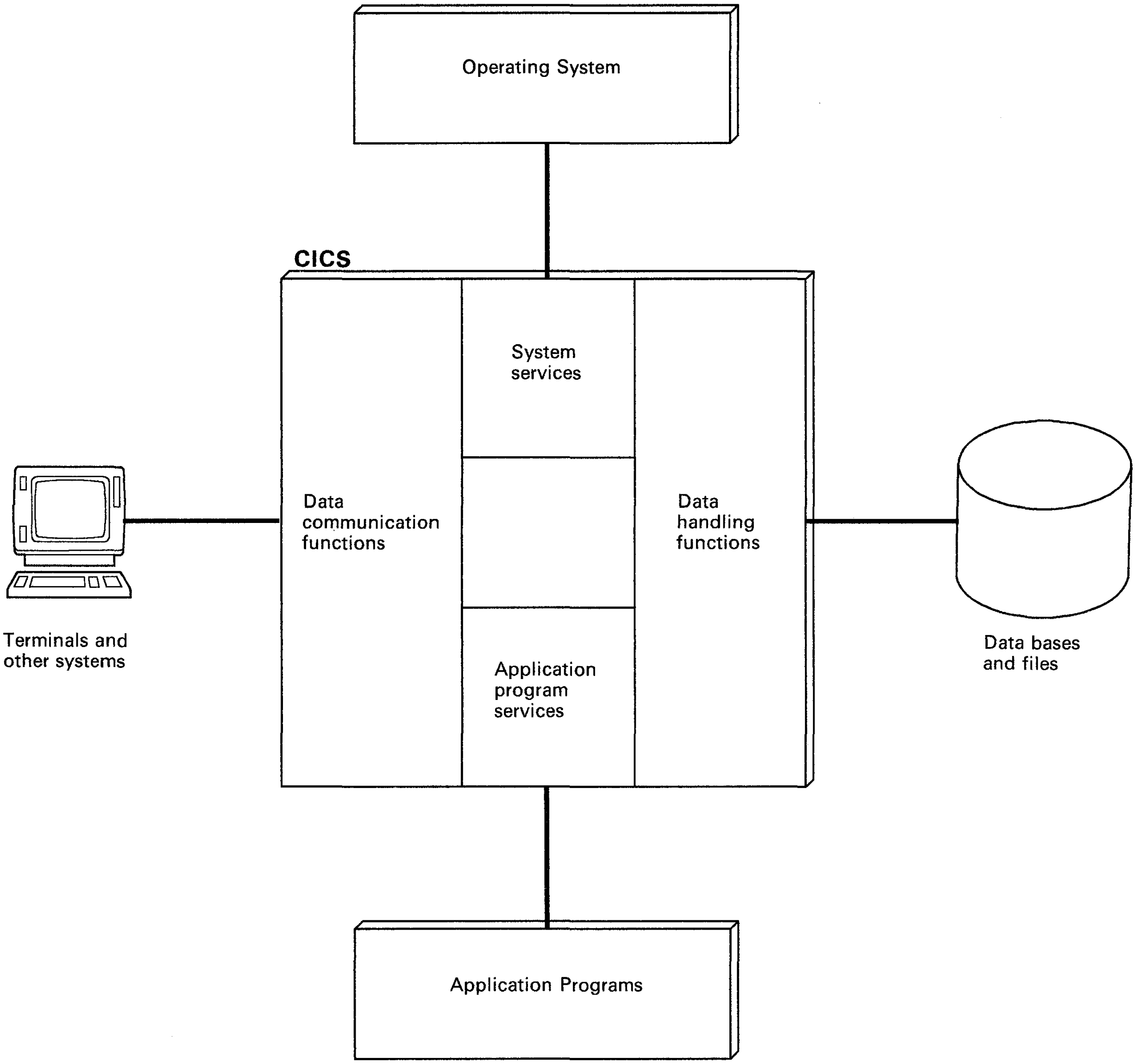


Figure 8. CICS and its Software Environment

Inside CICS

What Goes on Inside CICS?

You can write your own applications or buy ready-written application program packages. Either way, the people using CICS have online access to their data through these application programs running under CICS. They may also send messages to other CICS users.

In addition, application programs running under CICS can "talk" to other programs running within the same CICS system. They can also talk to programs running elsewhere in the same computer system, or (given the appropriate network support) to other computing systems.

After an initial look at the flow of control during a CICS transaction, we'll look at internal CICS facilities under four main headings. This part covers each separately:

- **Data communication functions** provide an interface between CICS and local or remote terminals to make the input or output of data easier. They provide a degree of device independence and format independence for application programs. There are also intersystem communication (ISC) and multiregion operation (MRO) facilities.
- **Data handling functions** provide an interface between CICS and stored data. They allow the data to be read or updated, while preventing unauthorized access and protecting the data from corruption. CICS contains interfaces to data base products and to standard file access methods. CICS also has routines to handle queues and scratchpad data used within CICS itself.
- **Application program services** can be used by your application programmers when writing and testing application programs. They provide an interface between CICS and the application program.
- **System services** provide an interface between CICS and the operating system. They include functions to control CICS, and to share resources.

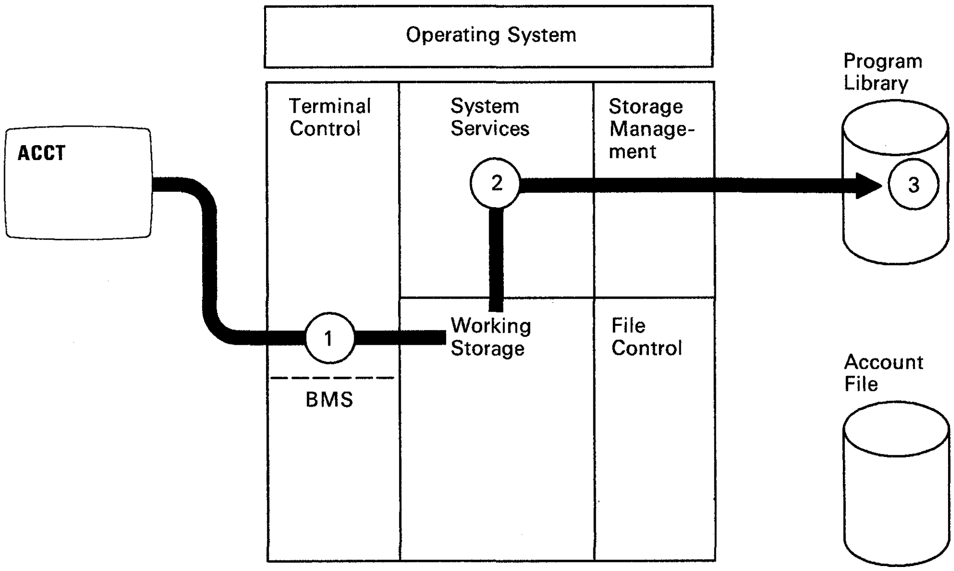
Other program products interface with CICS to provide additional facilities, such as data base management.

In the time it takes to process one transaction, CICS may receive messages from several terminals. For each message, CICS loads the application program (if it isn't already loaded), and starts a task to execute it. Thus multiple CICS tasks can be running concurrently.

CICS maintains a separate thread of control for each task. When, for example, one task is waiting to read a disk file, or to get a response from a terminal, CICS is able to give control to another task. Tasks are managed by the CICS **task control** program; the management of multiple tasks is called **multitasking**.

CICS manages both multitasking and requests from the tasks themselves for CICS (or operating system) services. So CICS processing continues while a task is waiting for the operating system to complete a request on its behalf. Each transaction being managed by CICS gets control of the processor when that transaction has the highest priority of those that are ready to run.

While it runs, an application program requests various CICS facilities to handle message transmissions between it and the terminal, and to handle any necessary file accesses. When the application is complete, CICS returns the terminal to a standby state. Figure 9 should help you understand what goes on.



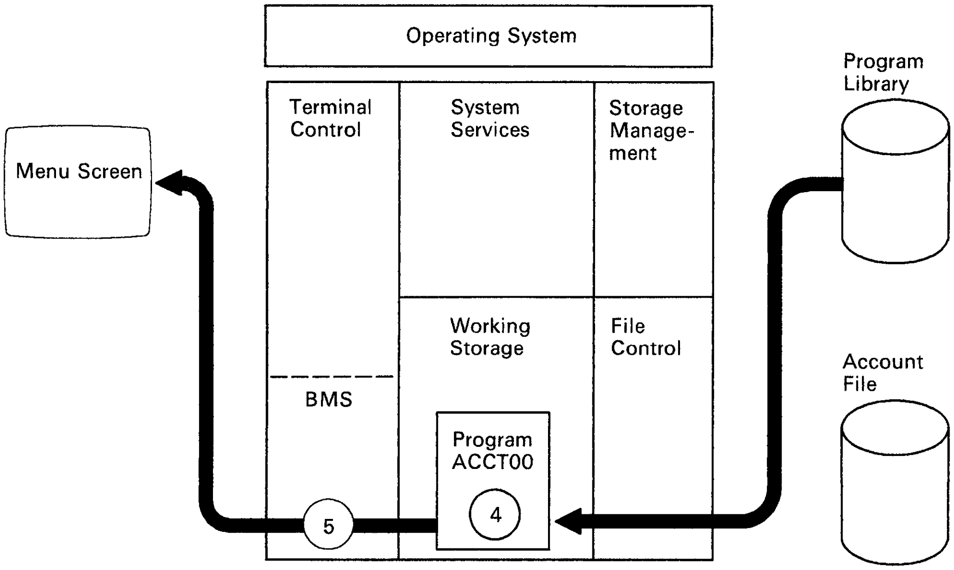
The flow of control during a transaction (code ACCT) is shown by the sequence of numbers 1 to 8 on the panels. The meanings of the eight stages are:

1. **Terminal control** accepts characters ACCT, typed at the terminal, and puts them in working storage.

2. **System services** interpret the code ACCT as a call for an application program called ACCT00. If the terminal operator has authority to invoke this program it is either found already in storage or loaded from...

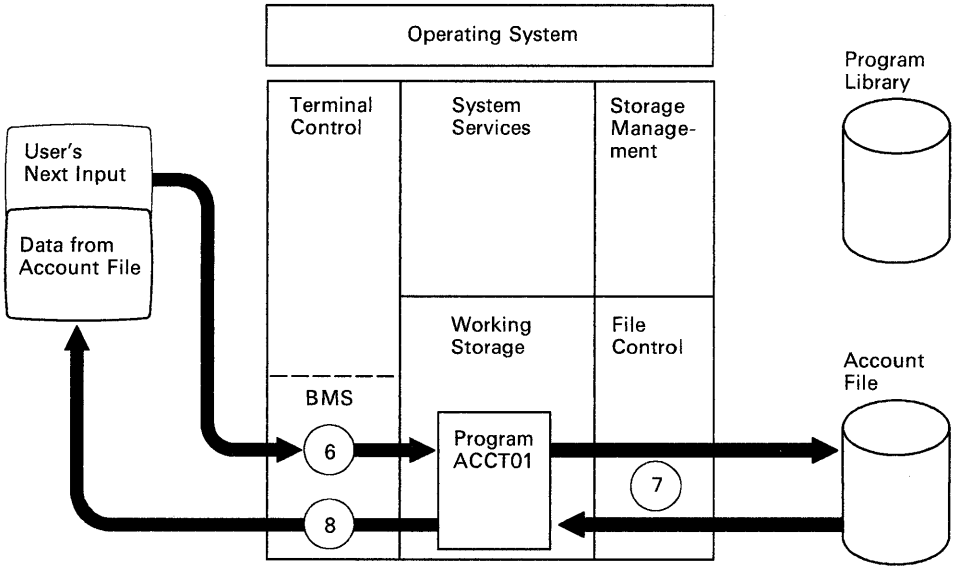
3. **The program library** into working storage, where...

Figure 9 (Part 1 of 2). The Flow of Control During a Transaction



4. A task is created. Program **ACCT00** is given control on its behalf. This particular program invokes...

5. **Basic mapping support (BMS)** and terminal control to send a menu to the terminal, allowing the user to specify precisely what information is needed.



6. **BMS** and terminal control also handle the user's next input, returning it to ACCT01 (the program designated by ACCT00 to handle the next response from the terminal) which then invokes...

7. **File control** to read the appropriate file for the information the terminal user has requested. Finally, ACCT01 invokes...

8. **BMS** and terminal control to format the retrieved data and present it on the terminal.

Figure 9 (Part 2 of 2). The Flow of Control During a Transaction

The transaction continues to run until it reaches a place in the program at which it's waiting for some activity (such as a disk access) to end. At this point, CICS gives the processor to the next task that can run. Of course, although it's the operating system that decides what is to run and when, control stays with CICS until a higher priority address space or partition (for example, VTAM) needs servicing. This allows CICS to maintain the priority of online working over batch work elsewhere in the system.

In this way, CICS controls the overall flow of your online system.

Besides doing all the transaction processing, CICS also supports the bookkeeping side of the system, by accumulating performance statistics and monitoring the resources used. This gives you the information that enables your user departments to be charged accordingly. It also allows you to find out which parts of CICS are being heavily or lightly used. This will help your systems people change the CICS setup when you wish to tune your system to improve its performance.

So much for the way in which CICS handles an individual transaction.

Data Communication Functions

CICS and Data Communication

This section discusses the various data communication functions and facilities that CICS provides.

Programs involving terminals normally require the programmer to have a detailed knowledge of the terminals' characteristics, and also the telecommunications line protocols. To make things worse, each program tends to be specific to the terminal running it. If you're using several different kinds of terminal, and if you want to be able to change terminals occasionally, such device dependence is inconvenient and time-consuming.

CICS **terminal control** and **basic mapping support (BMS)** facilities make it easier to write application programs that use terminals of different types. With BMS, you can use the same application program with different terminals, while using the full range of facilities that each terminal provides.

The section ends with a look at **intersystem communication, multiregion operation**, and **batch data interchange** facilities.

Terminal Control

Without products like CICS, application programmers would need detailed knowledge both of the characteristics of the terminals and the way they are connected to the host computer system. They would also need to extend that knowledge as the host system or the terminal network grew, and — worse — they would be forced to change their programs accordingly.

With CICS, a **terminal control table** (TCT) defines the devices that are attached to the system. Once the TCT has been set up, CICS will ensure that the terminals defined in it can send and receive messages from the system. Depending on the terminals attached, this may involve interfacing to telecommunications access methods such as VTAM.

The TCT is set up initially when the system is generated but can be changed, without the need for changing application programs, as the terminal network evolves. You can have several TCTs, identified by suffix characters, for use in different versions of your CICS system.

CICS/OS/VS Version 1 Release 7 allows you to make online changes to the TCT while your system is up and running. It also has autoinstall facilities that allow certain types of terminal to be defined to CICS automatically.

CICS terminal control facilities let you connect a wide variety of terminals and terminal subsystems to the host system. You'll find a list of terminals and access methods in Appendix B, "IBM Terminals and Subsystems Supported" on page 73. More detail appears in the *CICS/OS/VS Facilities and Planning Guide.*

Basic Mapping Support (BMS)

Basic Mapping Support (BMS) interfaces between the terminal control functions and the application program to provide **device independence** and **format independence**. It is available for most terminals supported by CICS.

BMS controls the way information is displayed at the terminal, holding in a **map** the relationship between the format of the data on the terminal and its identification in the program. The application designer can define BMS maps using CICS-supplied macro instructions, or alternatively an online map definition product such as SDF/CICS. (See "Online Development of Screen Formats" on page 37.)

In essence, then, BMS lets programmers deal with a symbolic screen layout rather than a potentially confusing and device-dependent serial stream of data.

A BMS map can include four things:

- The **format** of input and output data
- **Fixed data**, such as prompting messages, or page headings and trailer text
- **Symbolic names** of variable data fields within the format
- The **device type** to which the map relates.

With BMS, you can format the same data differently for different terminal types, emphasizing appropriate aspects of the data, for example by highlighting it. Such flexibility also allows you to exploit the special features of terminals such as color displays, while still allowing more limited terminals to run the same program.

By supplying a map for each kind of terminal that may need to use the program, you can often provide just one program without the need to constrain it to a particular type of terminal. This means, of course, that when a new kind of terminal is introduced it's a simple matter to add an extra map: the program is unchanged.

Have a look at Figure 10 on page 25.

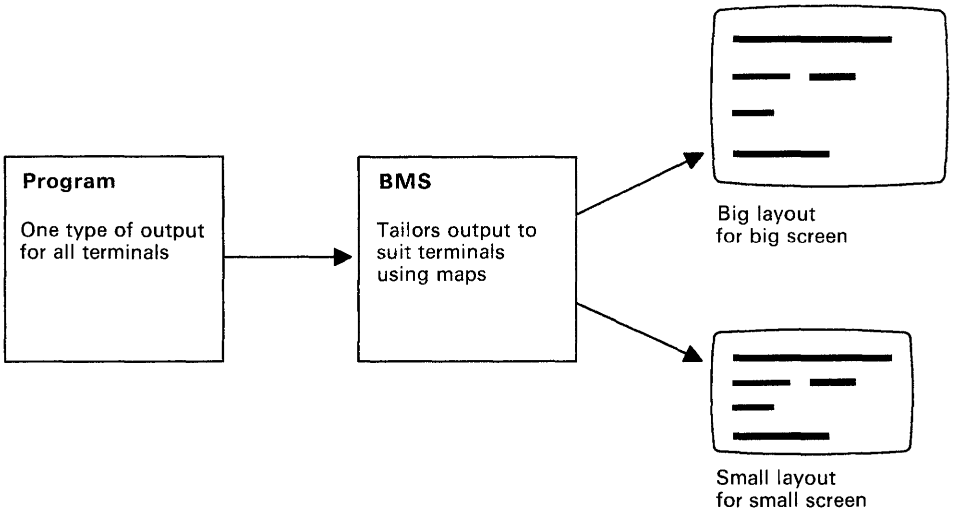

Program
One type of output for all terminals
BMS
Tailors output to suit terminals using maps
Big layout for big screen
Small layout for small screen

Figure 10. The Way BMS Interfaces between Terminal and Data

BMS gives symbolic access to input and output data fields without the use of device dependent code. It allows the program to send or receive data from a terminal regardless of the data's position on the screen. BMS translates terminal input data streams and presents them to the application program in a standard form, regardless of which type of terminal sent the input. And BMS converts output data to a form suitable for transmission to the terminal that is to receive it.

All this is done by maps defining and naming input or output fields. Output maps can contain field default values to specify titles, headers, masks, and so on.

BMS in Use

You define maps and map sets with BMS macro instructions. A map set is a group of related map definitions; you use it to create:

- A set of maps that are all normally used by the same application program.
- A set of physical maps for each device type, containing one map for each screen layout that the application needs to associate with that device.
- A set of symbolic description maps, which represent the application program's view of variable fields in the map.

The matching physical maps and symbolic descriptor maps are both generated from the same mapset source.

BMS also offers **terminal paging** and **message routing** facilities. An application programmer can create and transmit output messages that are larger than can be physically displayed at the terminal. BMS commands are available that allow the CICS terminal operator to retrieve "pages" of a message in any order. They can be read in the order they were prepared or by skipping forward or backward through the output pages.

The message routing facility, which can also use maps, lets an application program send output messages to one or more specific terminals that are **not** connected to that program's task. You can have the messages delivered either immediately, or at a particular time. This is particularly useful for transactions that need to route output to a printer.

BMS provides a standard transaction that allows one CICS terminal operator to send a text message to one or more colleagues.

Intercommunication

The CICS intercommunication facilities allow two or more systems or regions to communicate and/or share terminals and other resources. The two modes of intercommunication are **intersystem communication** and **multiregion operation**.

ISC and MRO have three main functions:

- **CICS function request shipping,** which allows command-level application programs (or programs written for the DL/I interface) in one CICS system to address resources in any connected CICS system or region

- **Distributed transaction processing**, which allows communication between transactions executing in different systems. So, for example, an application can be written as complementary transactions, each executing in an MRO connected region

- **Transaction routing**, which allows operators of terminals in one CICS region to run transactions in any connected CICS region. There's also a CICS-provided routing transaction.

The *Intercommunication Facilities Guide* presents a complete treatment of intersystem communication and multiregion operation.

Intersystem Communication

Intersystem communication (ISC) allows you to connect one CICS system to other CICS or IMS/VS systems in the same or other locations by means of Systems Network Architecture (SNA) facilities. What advantage is there in doing this? Well, it enables an application running in one CICS system to access files and data bases, and start processing, on connected systems.

Thus your installations can effectively distribute system resources (such as application programs, files, and data bases) throughout a network, rather than attaching locally required resources to the particular system requesting them.

It's worth noting that ISC works between CICS systems and, indeed, between CICS/DOS/VS and CICS/OS/VS systems, even if they are at different release levels. Figure 11 shows a typical intersystem communication configuration.

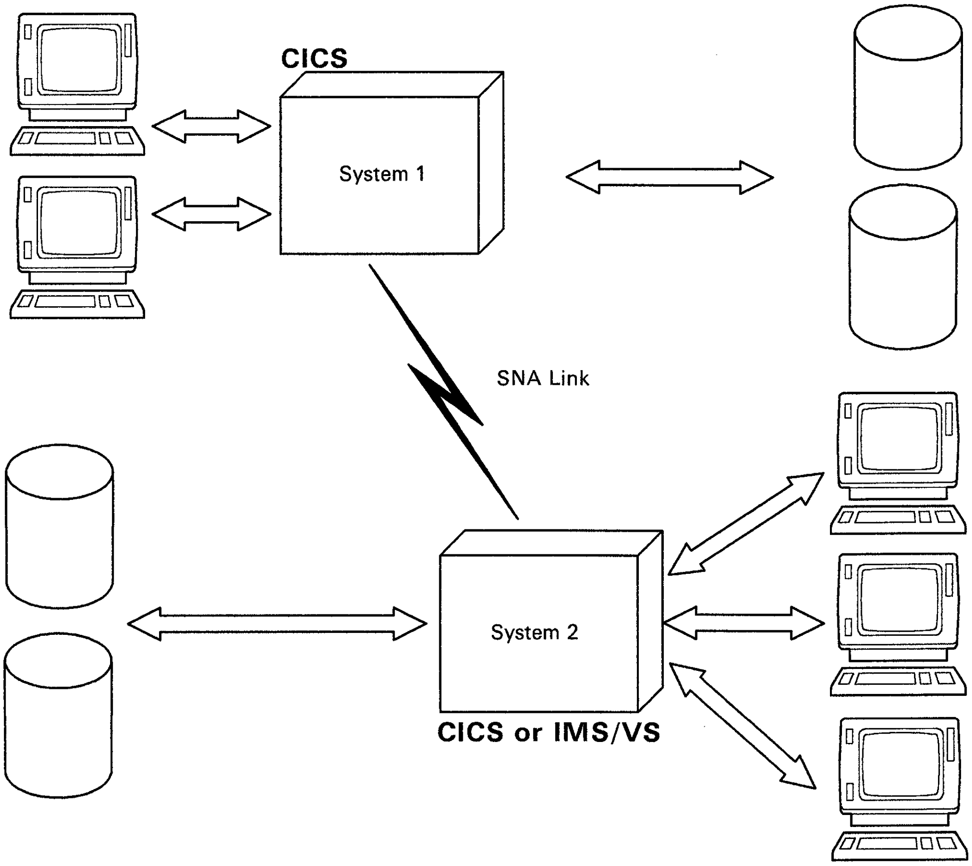


Figure 11. An ISC Configuration

Multiregion Operation

Multiregion operation (MRO) allows multiple CICS systems within the **same** processor complex to communicate with one another. This doesn't involve SNA networking facilities. The connected systems normally look like a single system to the terminal user. And there isn't normally any need to change application programs to use MRO. Figure 12 shows a typical multiregion operation system, with two CICS systems sharing access to one data base.

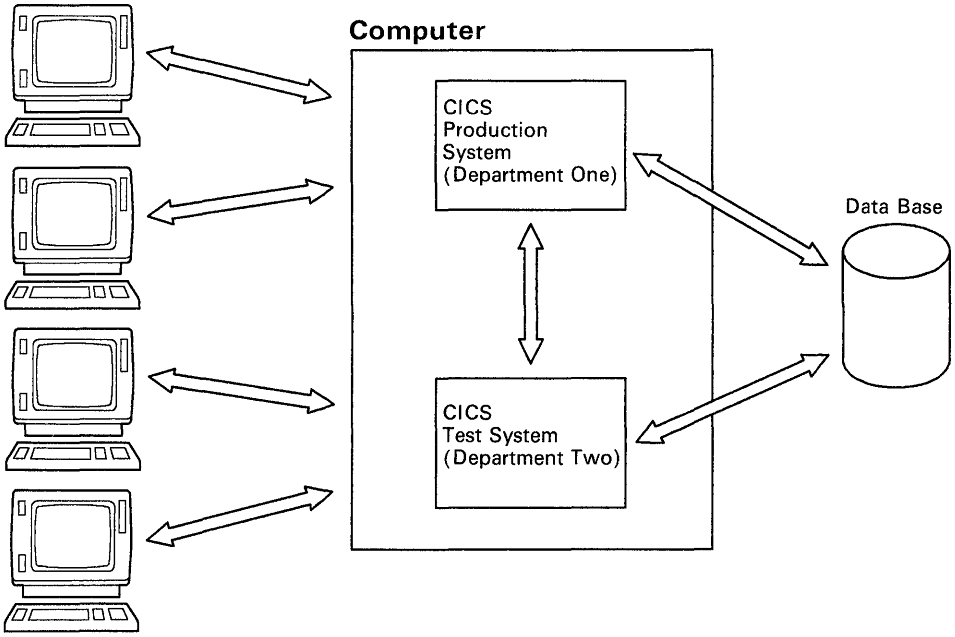


Figure 12. An MRO Configuration

Multiregion operation has several other advantages. For example, you can test new CICS application programs concurrently with your production operation of CICS, using a separate test region. This improves integrity and minimizes any impact on production while allowing selective access to existing resources and without requiring terminals to be dedicated to the test region.

Benefits of Intercommunication

These intercommunication facilities have a number of potential uses. You can have one or more CICS regions, with each region running one or more applications. For example:

- As already mentioned, you can test new CICS application programs in a separate test region concurrently with the production operation of CICS.

- You can exploit the advantages of an attached processor, multiprocessor, or dyadic system by running CICS in multiple MVS/370 or MVS/XA address spaces, to help balance the load on the total system.

- You can divide a CICS system on a departmental basis, providing independent operation for one department while allowing controlled access to resources belonging to another department. See "Isolating Sensitive Applications" on page 59.

- In more complex environments, MRO is a way of offering virtual storage constraint relief (VSCR). The system programmer can assign transactions and data resources to any connected system, again to help balance the load and get the best performance, particularly on dyadic processor systems.

- You can allocate CICS applications to different regions according to their response time needs. For example, you could run a data entry application, needing very rapid responses, in the highest-priority region and have an online data inquiry and update application running in a lower-priority region. Taking into account the longer periods of operator thinking time between terminal transactions, the inquiry and update users should still maintain satisfactory response times.

  You can move any non-CICS applications that need a great deal of processor time and few terminal interactions into a third region where they don't degrade the response times of the higher-priority transactions.

The system programmer specifies the regions that are associated with particular terminals, transactions, and resources. So you can optimize the total system across the CICS regions.

Communication with Other CICS and Non-CICS Systems

Many installations have a host computer and data base at a central location, linked to satellite computers. These maintain some local records and process local transactions at other locations, such as company branch offices. The local subsystems do not necessarily contain CICS, of course, but they communicate with the CICS region in the host system.

CICS has **batch data interchange** functions that enable applications to read in and write out batch data to named data sets in programmable subsystems such as the IBM 3790 Communication System or the IBM 8100 Information System. See Figure 13.

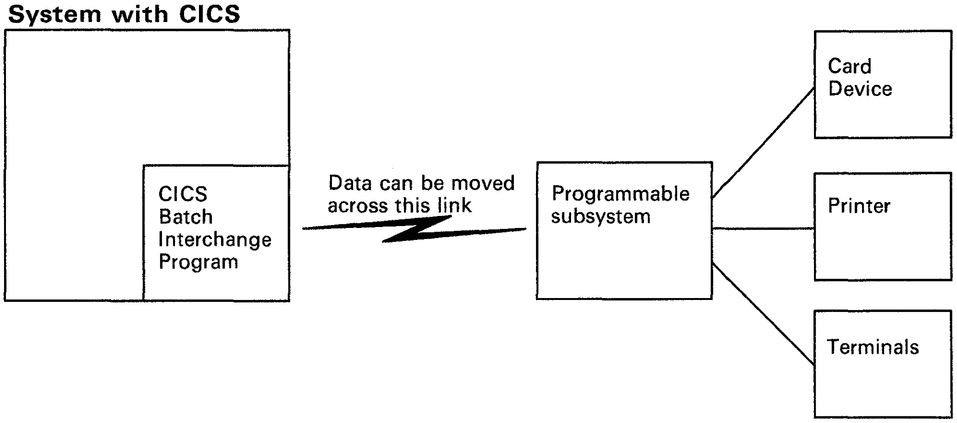


Figure 13. CICS Batch Data Interchange

Data Handling Functions

CICS Data Handling

This section discusses the CICS interfaces to **data storage** you can use, including **data bases**, **standard data sets**, and **areas within CICS itself**.

It also describes the features that help to preserve **data integrity** both in normal operation and under error conditions.

Data Storage

Interfaces within CICS allow access to data held in a variety of places, including:

- Data bases
- Standard operating system data sets
- Areas within CICS itself.

Data Bases

Data bases give the greatest degree of data independence. They can be shared between, accessed, and updated by batch programs as well as by CICS transactions, with full data base integrity.

IBM currently offers **hierarchical** and **relational** data base products for use with CICS. In a hierarchical data base (for example, DL/I or IMS/VS), the programmer has a logical view of the data base as a hierarchy of segments. Figure 14 shows the structure of a DL/I data base.

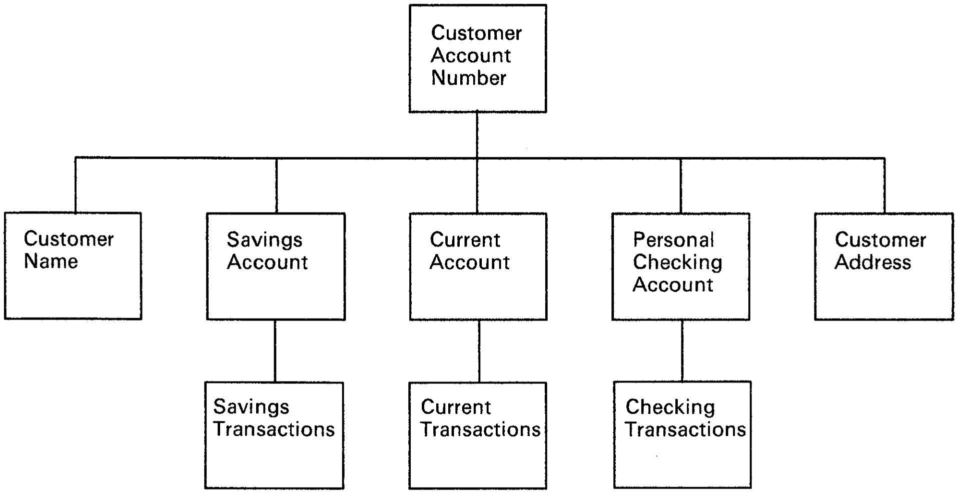
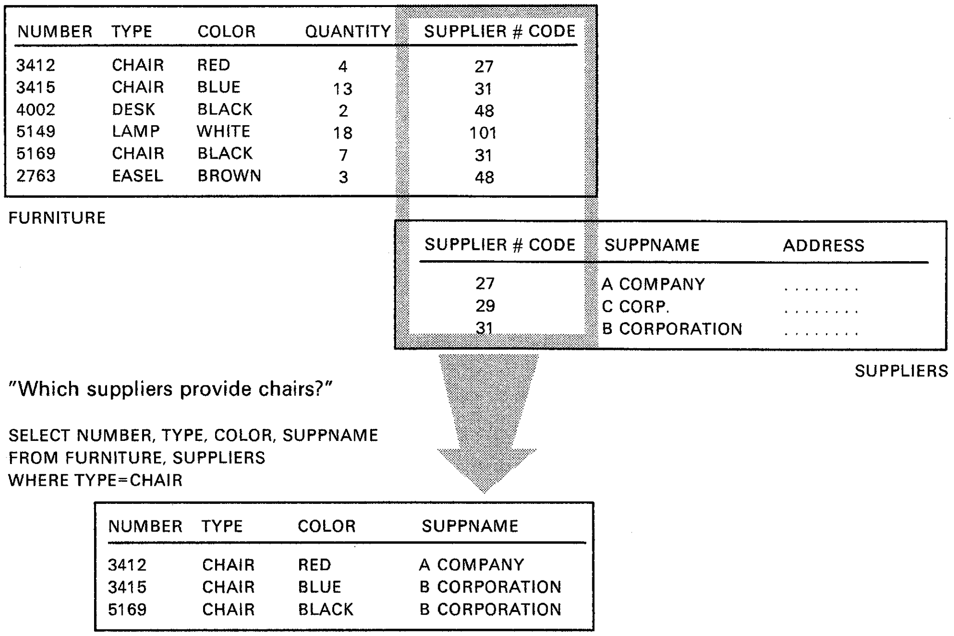


Figure 14. A DL/I Data Base

In a relational data base, such as IBM's Structured Query Language/Data System (SQL/DS) and Database 2 (DB2) program products, the data is presented to the user as a series of related tables. Figure 15 shows how an SQL/DS data base might be used.



NUMBER	TYPE	COLOR	QUANTITY	SUPPLIER # CODE
3412	CHAIR	RED	4	27
3415	CHAIR	BLUE	13	31
4002	DESK	BLACK	2	48
5149	LAMP	WHITE	18	101
5169	CHAIR	BLACK	7	31
2763	EASEL	BROWN	3	48

FURNITURE

SUPPLIER # CODE	SUPPNAME	ADDRESS
27	A COMPANY	....
29	C CORP.	....
31	B CORPORATION	....

SUPPLIERS

"Which suppliers provide chairs?"

```
SELECT NUMBER, TYPE, COLOR, SUPPNAME
FROM FURNITURE, SUPPLIERS
WHERE TYPE=CHAIR
```

NUMBER	TYPE	COLOR	SUPPNAME
3412	CHAIR	RED	A COMPANY
3415	CHAIR	BLUE	B CORPORATION
5169	CHAIR	BLACK	B CORPORATION

Figure 15. An SQL/DS Data Base

Either approach makes it possible to avoid duplicating data needed by several different applications. Each application can access the parts of the data base it needs without being concerned with the overall organization of the data base.

Application programs access DL/I data bases by means of a command level DL/I interface very much like the EXEC CICS... interface. In other words, programmers can code commands of the form "EXEC DLI..." in their application programs to access the data base. This has a flexible syntax, and offers EDF support — see "Execution Diagnostic Facility (EDF)" on page 36.

CICS/DOS/VS **assembler** programs are restricted to the earlier DL/I call interface.

CICS/DOS/VS supports the relational data base manager SQL/DS. CICS/DOS/VS application programs can contain both EXEC DLI and SQL commands.

Similarly, CICS/OS/VS supports the relational data base manager system program product DB2. And CICS/OS/VS applications that process DB2 tables can still access DL/I data bases.

IMS/VS Datasharing — first introduced in CICS/OS/VS 1.6.1 with IMS/VS 1.3 — allows more than one system to have concurrent access to a DL/I data base. This means concurrently active batch regions can share DL/I data bases with CICS transactions. There is similar support on DOS DL/I.

Standard Data Sets

Standard data sets offer a limited "within CICS" form of data base facility, with information organized in individual files. These are processed by CICS **file control** interfacing with the appropriate access method supported by the operating system. File management systems that CICS can handle include Virtual Storage Access Method (VSAM), and Basic Direct Access Method (BDAM). Broadly speaking, BDAM is slower than VSAM, but needs less real storage. However, BDAM cannot access files on a fixed block architecture device.

The two Data Facilities Products, DFP/370 and DFP/XA, include suitable versions of VSAM and BDAM. If you need support for VSAM buffers or VSAM record management code above the 16-megabyte (Mb) boundary on an MVS/XA system, you'll need DFP/XA 1.1 or later.

Amongst other things, file control provides command level access to browsing facilities. It also looks after buffer and block management, and deals with access method dependencies. This gives the application programmer a degree of **access method independence**.

Areas within CICS

Two other forms of CICS storage are **transient data** and **temporary storage**. Transient data handles queues of data to be sent to a terminal, such as a printer, and sequential files. Temporary storage provides an internal scratchpad facility for CICS's own use and for passing parameters between transactions.

Data Integrity

How do you protect your data from accidental loss or corruption? In other words, what can you do about maintaining the **integrity** of your data:

- During updating?
- If transactions are canceled?
- Following a system failure?

Maintaining the Integrity of Data During Updating

Where many users have access to the same data, there is always a possibility that several of them will try to change the same record at the same time. CICS prevents this by ensuring that one operator's updating of data is complete before another's may start.

Maintaining the Integrity of Data if Transactions are Canceled

During normal execution of a task, CICS saves information about all protected data that is being changed, putting this information in the dynamic log (an area of storage allocated to the transaction as required). Protected data covers, amongst other things, any files you nominate, and data bases. CICS deletes this log information after the successful completion of the task. However, if for any reason the task is not completed, the data changes can be reversed.

CICS provides a **dynamic transaction backout (DTB)** facility to manage this. If a transaction fails, due perhaps to application program error, data access error, transmission error, or because an operator decides to cancel a transaction, DTB reverses the updates that have been carried out by the transaction involved.

The process of canceling changes in data works backwards from the last change before the failure, hence the name dynamic transaction backout. The backout occurs within the same task. This safeguards other tasks from the possibility of using corrupted data, because modified data is not released for use by them ("committed") until the current task has finished with it.

Application programs can specify intermediate **synchronization points** (sync points). These are points at which data updates or modifications are logically complete. Sync points delimit a **logical unit of work** (LUW). In the event of failure, a sync point tells CICS that changes made before that point (that is, during a preceding LUW) do not need to be backed out. Sync points help to speed up and simplify recovery from failure in a long-running task. However, as detailed in the *Recovery and Restart Guide,* we recommend the use of short tasks that need no sync points.

The Subsequent Recovery: If you use DTB, it can be followed by a **restart** function which allows the transaction to be retried immediately.

Transaction restart is an optional facility that allows a canceled transaction to be restarted automatically without intervention by the terminal operator, provided that certain conditions are met. See the *Recovery and Restart Guide* for details. The facility allows the application program to perform additional recovery functions written by the programmer.

If an application program runs into problems, it can choose to call the **rollback** facility to cancel the changes it has made in all recoverable resources during the current transaction. These are then restored to the state prevailing at the previous sync point.

These transaction recovery functions are usually transparent to the terminal user.

Maintaining the Integrity of Data Following System Failure

In addition to the dynamic log mentioned above, CICS keeps track of the activities of programs in the system through system logging and automatic journaling. A journal is a sequential data set (on tape or a direct access storage device) which can be used to provide, for example, a record of operator and system actions, or a means of recovering superseded data. You can specify the extent of this journaling by what you put in control tables set up when the CICS system is generated.

In the event of a system failure, CICS provides recovery programs to rebuild the system, using the system log.

CICS will try to keep running even though a fault has occurred. However, drastic faults (such as a loss of power or an operating system failure) will cause CICS to stop. The logging and journaling facilities enable system restarts to be carried out.

When CICS is started, it can find out whether the system terminated “cleanly” or was interrupted by a system failure. In the latter case, CICS will automatically perform an **emergency restart**. This uses the system log information recorded during the previous execution of the system to restore CICS to a predefined logical point which existed before the failure.

If the system terminated cleanly, CICS performs a **warm** start. This initializes the system exactly as specified in the chosen System Initialization Table (SIT), without reference to any previous run of CICS.

Application Program Services

CICS Application Programming

You may choose to use one or more complete application system packages which can be tailored to your particular requirements. Many of these are available from IBM or other sources. Additionally, or alternatively, your installation can develop its own application programs, using services provided by CICS itself.

This section deals with the **programming languages** that you can use for this purpose and describes the facilities provided for **program development under CICS**.

Programming Languages

Your application programs can be written in COBOL, PL/I, RPGII (under VSE only), or assembler language. For each of these languages, programmers include CICS commands within the program to use the particular CICS facility required.

The facilities that must be provided for the application programs will vary according to the environment within which CICS is running. For instance, simple data-handling functions are provided by CICS itself, whereas operations on more comprehensive data structures are made easier by using data base products such as DL/I and SQL/DS.

Application Program Development under CICS

CICS systems provide various kinds of assistance for program development:

- Facilities within CICS; for example,
  - A command language translator which runs as a batch program and preprocesses source code CICS commands into the form that CICS requires
  - A diagnostic aid (the Execution Diagnostic Facility, EDF) to enable the programmer to test and debug application programs online
  - A command-level interpreter for online syntax checking and, optionally, command execution
  - Extensive tracing and formatted dump facilities.
- Associated program products which run under CICS; for example,
  - A program that permits easy interactive development of terminal screen formats (Screen Definition Facility/CICS).
  - Application program generators such as Cross System Product (CSP), and ELIAS.

The normal program development method is to design and code a program, translate and compile it, and then test it. Because, very often, CICS is in daily use helping you run your business, it's important that you don't install new applications on to your production CICS system until you're sure they are reliable and bug-free. Testing is usually done with EDF, and in a test CICS system or in an MRO environment separate from your key applications. See "Isolating Sensitive Applications" on page 59.

You'll find that the *CICS/VS Application Programming Primer*, (SC33-0139), describes the entire application development cycle.

Before we look at the various program development aids, glance at Figure 16 for an overall view of the application program development process.

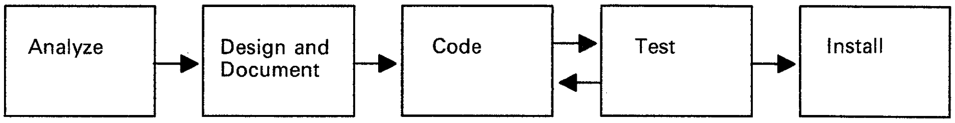


Figure 16. The Application Program Development Process

Any modifications during this process will feed back from later to earlier stages.

Command Language Translator

The command language translator runs as an offline batch process. It accepts as its input your source application program which will, of course, contain CICS commands. The translator's output is an equivalent source program, but with the CICS commands turned into statements in the language of the source program. It also provides diagnostic information for any syntax errors it finds.

At execution time, these statements invoke the EXEC interface program, which gets the information it needs from parameters passed, and passes control to the appropriate CICS facility.

The translator also processes commands to access DL/I data bases.

Execution Diagnostic Facility (EDF)

The Execution Diagnostic Facility (a CICS-supplied transaction called CEDF) is designed to allow an application program to be tested and debugged online.

With EDF generated and the terminal in DEBUG mode, you can have every CICS command intercepted and interpreted, prior to its execution. Each command is displayed in source form along with keywords, keyword arguments, and specified options.

You can check the displayed information for errors, and to confirm that it is as intended. You can correct, or modify, any argument values and execute the modified command.

After execution, but before control gets back to the application program, the command is again intercepted and displayed. So now you can check the response and any returned values generated by CICS.

You can even overwrite these responses. This lets you force the application program to invoke its exception condition routines, without having to generate the particular condition you're interested in, or having to create special-case test data.

Command Interpreter

The command interpreter (a CICS-supplied transaction called CECI) provides you with an online interactive system. For example, you can construct on a screen an EXEC CICS command of any variety. The interpreter helps you with syntax, the required operands, defaults, and various options for the selected command, allowing you to build that command. You can validate the command, and (if you wish) in most cases proceed to execute it.

CECI is there to help programmers, whatever their level of experience. It provides virtually the same options and diagnostics as the batch translator. When you execute a command, it's as though a real program had run, with the result of the command being intercepted for you both before and after its execution.

Online Development of Screen Formats

An important aspect of application program development is the definition of your screen formats, or maps. If you use BMS, you can either use the BMS macros (a batch process) or work online with a separate program product called Screen Definition Facility/CICS (SDF/CICS). This not only enables you to define BMS maps online, but also helps you convert your existing map definitions into SDF/CICS format.

Using SDF, you can build a screen layout online and automatically create and save the corresponding map and map set definitions in a map specification library. You can display these maps, edit them, and save them.

Here's a list of the main features and functions of SDF/CICS:

- Throughout an online session, you always have reference information available for both normal operation and error situations.
- You can assemble a collection of maps to form a page. And you can define, inspect, and test this page online.
- You can define a set of default values to apply for a given online session.
- Your map specification library and the associated library management functions give you a complete inventory of all your maps and map sets, with status information. You can delete, rename, or copy objects, and get map details printed out.

Application Program Generators

Next, we shall look at two application program generators. They are the Cross System Product (CSP), and the Entry Level Interactive Application System (ELIAS).

It is worth noting that some of our customers have installed CICS systems (plus one or more of these program products) solely for program development purposes.

Cross System Product: CSP simplifies the development of applications that use the 3270 series of terminals. With this one interactive tool, programmers can build applications, define screens and logic, test the application, and put it into production.

Cross System Product/Application Development (CSP/AD) uses prompts, menus, "fill-in-the-blanks", and HELP screens to guide programmers through the development steps. By using the prototyping and interactive testing capabilities, you can place the eventual application into production earlier.

CSP/AD supports sequential, VSAM, and DL/I files. Data definitions are entered using menus and prompts, thus speeding up the process and increasing the accuracy. So the whole application development cycle is speeded up.

CSP/AD's output is a file that defines the application; Cross System Product/Application Execution (CSP/AE) then processes this file to execute the application.

There's a companion product: Cross System Product/Query (CSP/Q). You can use CSP/Q to query data files and create reports either by using step-by-step menus or with a command language.

ELIAS: ELIAS is an interactive product available to VSE users. It provides interactive dialogs which will help you:

- Define DL/I DOS/VS data bases.
- Backup, recover, and reorganize DL/I DOS/VS data bases.
- Construct batch or online applications using COBOL or PL/I. ELIAS/1 Release 2 can use all types of VSE/VSAM files in these programs.
- Provide user documentation.
- Define CICS screens.

Trace and Dump Facilities

Finally, for the more complex problems that involve system interactions, CICS offers the **trace control** program. This is an optional CICS debugging aid which you can use to trace the path of an application program. Trace control consists of the trace control program itself (which is invoked before and after each CICS command) and a trace table. CICS programs can generate requests that:

- Select which CICS functions are recorded.

- Start and stop the recording of trace information about the application program in the trace table, on a CICS-wide basis.
- Enter user information in the trace table.

Dump Control allows you to request the writing-out of the contents of selected areas of storage to a sequential data set for printing. Dump control is automatically invoked when a transaction terminates abnormally.

The areas selected to be dumped can include main storage areas related to the requesting task; CICS control tables; and, particularly, the trace table itself for help in debugging.

There's a **formatted dump program** whose output contains a dump of the CICS partition or region. Each control block is printed separately in the dump, and is preceded by a heading; for some of the blocks the important fields are printed by name. CICS also produces a full range of diagnostics to describe execution-time errors.

You can also make the formatted dump program take a "snapshot" (that is, an unformatted dump) of the CICS region or partition before producing the formatted dump.

The formatted dump program is usually invoked if CICS terminates abnormally, or because the operator requests a dump at system shutdown.

System Services

CICS System Services

CICS provides a number of control functions. Three of these in particular supervise and control the allocation of resources within the system. These functions are: **task control, program control** and **storage control**.

Another control function (**interval control**) offers timer services.

Task Control

Task control keeps control of the status of all CICS tasks. Many of them will be being processed concurrently and task control allocates processor time among them. In CICS terms, this is called **multitasking**. You may prefer to think of it as multithreading.

When an operator requests a transaction, normally by logging on and keying in a transaction code, CICS checks the status of the operator and the terminal. This ensures that the operator is known to the system and that the transaction is valid for that user and that terminal. Task control then creates a task for the transaction.

CICS tries to give the best response times to the most important or urgent work. Usually, several tasks are competing for resources, so a transaction, an operator and a terminal are each assigned a priority related to the importance of the functions they carry out. CICS sums these priorities to give the overall priority of the task, and uses this priority to decide the order in which to process competing tasks.

Since transactions are not normally processed through to completion in a single, uninterrupted operation, CICS makes such decisions many times during the execution of a task. A transaction is processed up to an instruction involving input from a file or a terminal, for instance. Then, while the transaction waits for its input, another waiting task begins or resumes execution.

Program Control

As soon as task control has started a task for the transaction, it becomes program control's job to associate the task with the appropriate application program.

Although many tasks may need to use the same application program, program control loads only one copy of the code into the CICS partition. Each task threads its way through this code independently, so many terminal operators can all be entering and using transactions that are concurrently using the same physical copy of an application program. (However, this does not apply to programs written in RPGII.)

Storage Control

Within its region, CICS maintains full control over virtual storage. Storage control acquires, controls, and frees dynamic storage – this being the space left within a partition after CICS itself is loaded. Dynamic storage is used for programs, input/output areas, work spaces, and so on.

Timer Services

Interval control functions allow applications to start and control a range of time-dependent actions (such as starting a particular transaction at a certain time of day, signaling when a specified period has elapsed, and so on).

What's next?

In the next part of the book, we move back from inside CICS to tell you what hardware and software you need to get a CICS system up and running.

We also have a bit to say about the operational aspects.

Part Three

This Part of the book describes:

- The hardware and software needed to run CICS
- System design and configuration
- CICS operating procedures
- Monitoring and tuning
- Application design

Running a CICS System

Running CICS

This part summarizes the various options available when choosing the elements of a CICS system. Firstly, you need a suitable hardware configuration and operating system. Secondly, you need your suite of application programs to run under CICS and interact directly with your online users. Thirdly, you need the people to set up a CICS system, to monitor it and operate it, and to develop it over time.

For CICS/OS/VS 1.7 users, the *CICS/OS/VS Facilities and Planning Guide* is now available. This replaces the *System/Application Design Guide* and contains a wealth of useful information for your CICS planning stage.

In selecting a configuration, you will need to balance costs, flexibility and expansion capability, and benefits. Your choices must take into account both what you want the system to do and how much load it will be under during normal use. Usually, of course, your choices will also depend on the requirements of the system users.

This part is divided into five sections:

- The **hardware and software needed to run CICS**.
- **CICS system design and installation**. This gives advice about planning a new CICS system to meet your needs, and putting the design into practice.
- **Operating a CICS system**. This discusses basic operating procedures.
- **Monitoring and tuning** your CICS system. This looks at basic performance assessment.
- **Application design aspects**. This covers programming methods and data security.

Figure 17 offers a more detailed view of the software environment that "surrounds" CICS.

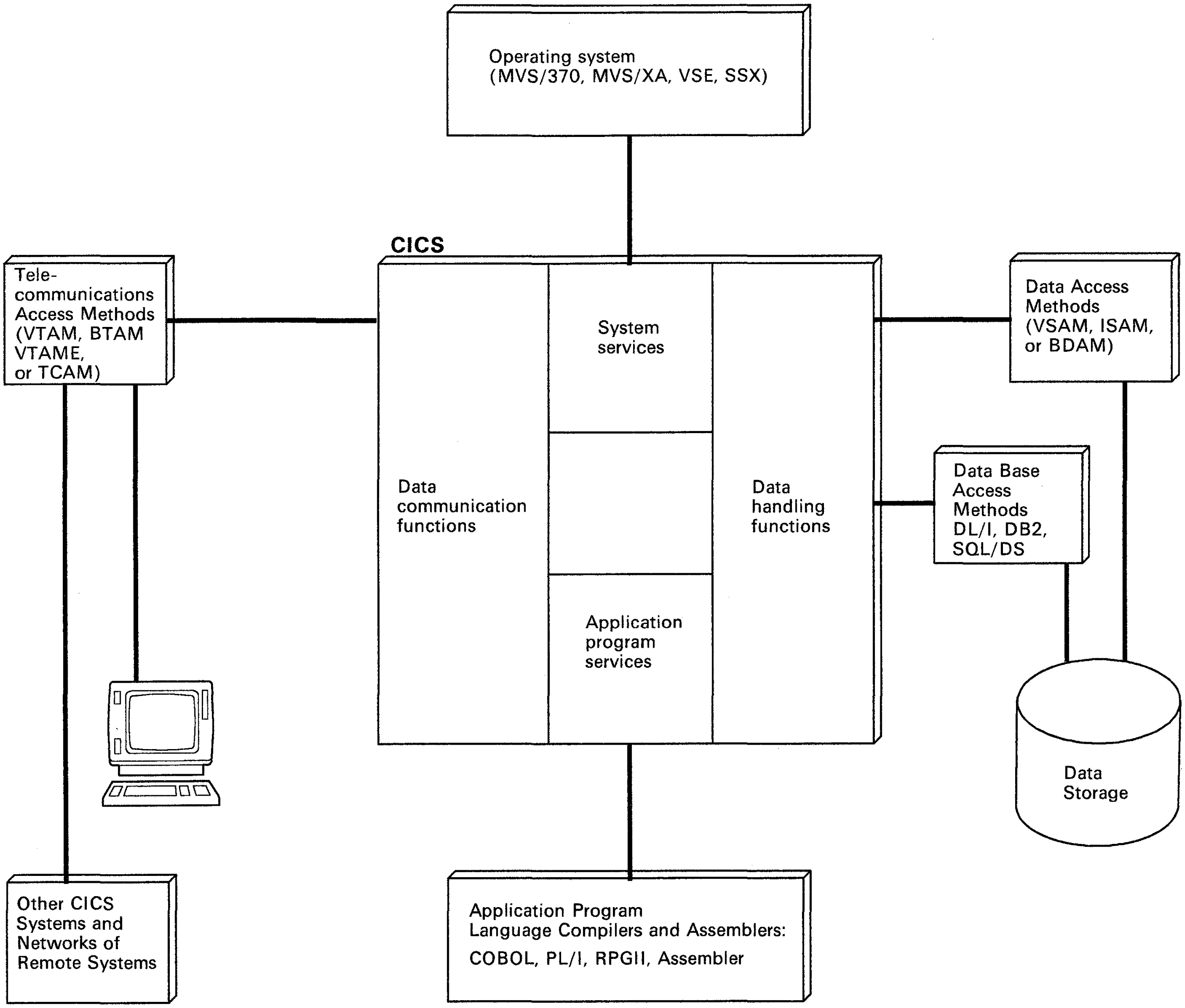


Figure 17. The Software Environment That Surrounds CICS

Hardware and Software Needed to Run CICS

CICS Hardware and Software Needs

This section briefly describes the requirements of a system on which CICS is to run, using the following headings:

- Processor, and related hardware.
- Operating system software.
- Data and data base access methods.
- Telecommunications access methods.
- Language compilers and assemblers.
- Other software.
- Storage capacity.

If you find all this information a bit difficult to absorb, you might prefer to see the relevant chapter of the appropriate *Installation and Operations Guide* for your operating system environment. For another source of information about the release levels of software supported, don't forget your local IBM Branch Office.

Processor and Related Hardware

You can run CICS on any central processor that supports one of the operating systems listed below. Suitable processors include the IBM System/370, 43xx, 303x, 308x, and 3090.

Your configuration must include enough input/output devices and direct access storage devices to handle your requirements for system output, system residence, and system data sets.

Operating System Software

CICS runs as one or more tasks under VSE or SSX as CICS/DOS/VS, or under MVS or MVS/XA as CICS/OS/VS. CICS/OS/VS continues to take advantage of various 31-bit addressing benefits under MVS/XA. CICS may be run in a real machine, or in a VM/370 environment.

The CICS interface to JES enables system programmers to access the system spool files maintained by JES2 and JES3 on MVS systems. This interface — new for CICS/OS/VS Version 1 Release 7 — allows you to exchange documents and files through a JES Remote Spooling Communications Subsystem (RSCS) network using EXEC CICS commands in command-level application programs.

Data and Data Base Access Methods

CICS supports VSAM and BDAM.

You need a current release of VSAM for system generation, and for certain options such as recovery and restart. CICS/OS/VS 1.7 requires you to have either DFP/370 or DFP/XA — these both include suitable versions of VSAM and BDAM.

CICS/DOS/VS 1.6 also supports ISAM.

CICS/OS/VS 1.7 supports IMS/VS DL/I and DB2 data base access methods.

CICS/DOS/VS 1.6 supports DOS DL/I and SQL/DS data base access methods.

Telecommunications Access Methods

The two access methods that are most frequently used are **Virtual Telecommunications Access Method (VTAM and VTAME)** and **Basic Telecommunications Access Method (BTAM).** MVS users have a third alternative: **Telecommunications Access Method (TCAM)**. Of these, VTAM is IBM's primary communications access method.

VTAM's ability to support programmable terminals and subsystems (such as the IBM 3600 Finance System) allows some of the work normally done in the processor to be offloaded. In some cases, a significant proportion of the work can be completely handled by the programmable terminal without the need for a transmission to the host processor.

Your choice of telecommunications access method depends on your terminal network which, in turn, will depend on your application needs.

Language Compilers and Assemblers

For CICS/OS/VS 1.7 you can use:

- COBOL/VS
- Full ANS COBOL V3, V4 Compiler and Library
- VS COBOL II
- OS PL/I Optimizing Compiler and Libraries
- Assembler XF or Assembler H V1 (MVS/370 systems)
- Assembler H V2 (MVS/XA systems)

For CICS/DOS/VS 1.6 you can use:

- Full ANS COBOL V3, V4 Compiler and Library
- DOS PL/I Optimizing Compiler and Libraries
- DOS/VS RPG II Compiler
- Assembler XF or Assembler H V1

Other Software

Other software includes:

- Graphics software, such as the Graphical Data Display Manager (GDDM) and Interactive Presentation Graphics (IPG).
- Application program development and generation aids (such as SDF/CICS, CSP, DISOSS, SMF, and, for MVS users only, RACF)
- Performance analysis software (such as CICSPARS, SLR Version 2).

See also Appendix A, "Some IBM Programs for Use with CICS" on page 65.

Storage Capacity

You must have enough main storage and direct access storage to satisfy the combined requirements of CICS, the host operating system, appropriate access methods, batch needs, and application programs running in the host. The amount of storage you need (both main storage and online DASD storage) depends on the configuration of the whole computer system on which CICS is to be installed. For example, it depends on the number of terminals with online access at any one time, on the telecommunication and operating system storage access methods, and so on.

Storage needs vary considerably, depending on your configuration, the system's usage, what functions you perform, and so on. The *Performance Guide* will help you in setting, measuring, and achieving your performance aims.

CICS System Design and Installation

Designing and Installing Your CICS System

This section gives advice about the **design strategy** of planning a CICS system and the **installation strategy** of putting the design into practice.

We discuss the alternative installation methods — to install CICS as a **pregenerated system**, or to **assemble system programs. Resource definition** is mentioned, along with the role of the system programmer in it.

We also touch upon the **System Installation Productivity Option** (SIPO) that you can use to help in the installation process.

CICS is a powerful and complex system which can contribute a great deal to providing your information needs. But you must be ready to invest effort in the planning process and to perform regular monitoring after CICS has been installed so that you can adjust the system as needed.

Design Strategy

You must design online systems to take account of the needs of their users and of other system applications. Whatever the scope of the proposed system, you must plan its recovery and restart aspects right from the start. To meet these aims, it may be a good idea to set up a CICS design group that includes people with a knowledge of application needs, installation requirements, and CICS facilities. This will certainly be necessary where a large, sophisticated CICS system is proposed. Smaller, more straightforward online systems won't require a formal design team.

(The first book you should go to after this one for considerably more information is the *CICS/OS/VS Facilities and Planning Guide.*)

The beauty of CICS is its flexibility. You can gradually expand your system from a simple handful of terminals on one, small, processor right up to vast, interconnected multiple CICS systems on larger, more powerful processors as your information needs change and develop.

System design is an iterative process, both during the initial design of the system and during any subsequent evolution it may undergo. Your designers must therefore be flexible in their approach and be prepared to make changes. However, they must also avoid the urge to continue to add improvements, particularly those oriented toward application design, to an extent that may prejudice implementation time scales.

It is important to choose a point at which to freeze the system design for the initial implementation. You can then keep notes of subsequent suggestions for improvement, to try out after the initial system has been installed.

Installation

A Checklist

Here's a checklist of the activities involved in the installation of CICS:

- Obtain the necessary CICS books — see page v.
- Install at least the minimum necessary machine configuration.
- Install the correct level of operating system.
- Plan and install the terminal network.
- Train your systems analysts and system programmers. They will then need to plan in detail the schedules and procedures for your conversion to an online environment.
- (Possibly) select one or more application packages that meet your information processing needs — see Appendix A, "Some IBM Programs for Use with CICS" on page 65.
- Train your application programmers, system operators, and end users.
- Install, and if necessary customize, your CICS system.
- Design and create your data base, or data sets.
- Design and create any BMS formats and maps you intend to use.
- Design and produce your application programs.
- Develop procedures to guard against accidental (or malicious) loss or misuse of data.
- Develop appropriate backup plans to protect vital applications.

Now let's look briefly at the main aspects of CICS installation. Your two major jobs are:

- Choosing your CICS configuration options
- Defining the CICS resources.

CICS Configuration Options

These days, most people simply use a pregenerated CICS system — either as supplied, or customized to their particular needs. End of story.

However, you can also generate the online system control programs with a variety of options — you can even generate any one program several times with different combinations of options. Subsequently, when CICS is initialized, you can select the required version of each system control program.

CICS has a modular structure. You can select options appropriate to your applications to help you achieve the best balance for your applications between system function and performance.

CICS Resource Definition

You have to define the DB/DC environment to CICS. You do so by means of a series of **tables**. These tables are organized so that there is one type of table associated with each **resource.** For example, you define files with the File Control table. CICS enables you to maintain several versions of some of the tables (as with system control programs) by using suffix characters. This allows you to maintain separate versions (for example) for peak loads, for testing, or for special shift work, in addition to the tables you use during normal operation. You then tailor the CICS system operationally by assembling the appropriate set of tables.

You prepare tables by assembling appropriate macro instructions. You can assemble new tables at any time before a CICS system initialization. Since CICS Version 1 Release 6, you've been able to add or change certain resource definitions when your CICS system is up and running. This facility is considerably extended for CICS/OS/VS Version 1 Release 7.

Look at the appropriate CICS *Resource Definition Guide* for more information.

The Role of the System Programmer

System programmers are able to debug CICS system problems and write CICS user exit routines and error programs. They are responsible for defining the resources in the CICS control tables, and setting up the master terminal operator's procedures. They must work with application designers and programmers to ensure that each new application is supported by any necessary system features and resources.

System Installation Productivity Options

You can install CICS as one of a set of software options using System Installation Productivity Options (IPOs). These are specifically designed to cut down the resource and effort needed from you to plan and create your CICS system. The System IPOs provide installation and verification procedures, installation JCL, sample programs, and pre-assembled CICS modules and tables.

There are a number of System IPOs for MVS, VSE, and SSX users. They have two main parts: the base, which contains the System Control Program, and a number of optional features.

For VSE/SP users, we have the SIPO/Extended packages (SIPO/E). These include in the base a program called the Interactive Productivity Facility. This has an interactive interface which helps you install CICS (and other products such as DL/I) by means of a series of screen displays.

Operating a CICS System

Operating CICS

This section of the book looks at the CICS **master terminal** and at the people involved in operating a CICS system. These are the system operators, and the master and supervisory terminal operators.

The CICS book to look at here is the *CICS/OS/VS CICS-Supplied Transactions.*

CICS Master Terminal

CICS is usually started from the main operating-system console. After that, CICS operating instructions are given from a terminal called the CICS **master terminal.**

The master terminal can be a separate terminal dedicated to this work or it can be a user terminal that runs application programs as well. Alternatively, the operating system console can also act as the CICS master terminal console.

The System Operators

System operators are responsible for the smooth running of the processor and of the system as a whole. They need to know how to start and stop the CICS system. They also must understand the various messages and codes that CICS issues. They must know what recovery action to take if the machine fails or if CICS terminates abnormally. And, of course, they need to be able to load tapes, disks, and other resources as required by CICS or by application programs.

The Master Terminal Operator

The master terminal operator is responsible for the smooth running of the CICS system. Master terminal operators must understand CICS master terminal tasks and procedures, the CICS network, how to control CICS files, journals, and so on. It is the master terminal operator who will activate CICS network elements on demand, review any performance bottlenecks, terminate looping or runaway transactions, and so on. Master terminal operators can, for example, allow further users to connect to a particular transaction.

The master terminal functions are available only to the master terminal operator, although anyone can be a master terminal operator unless the transaction is restricted. Master terminal operators can be people who have only this function, or they may have other operations functions as well. For example, the person who controls the system may also act as the CICS master terminal operator.

CICS provides an interactive interface for the master terminal operator. This permits flexibility in running CICS, and enables the master terminal operator to inquire about the allocation of resources and to change resources while the system is running.

The operator can use the monitoring information outlined in the section called "Statistics and Monitoring Facilities" on page 55 to adjust the system manually if appropriate. However, installations vary greatly. Some need to make manual adjustments, some don't. Systems are more likely to need monitoring and frequent adjustment if they are heavily loaded or if the load is very variable.

The choice of master terminal operator and master terminal depends partly on the need to have secure access and partly on the amount of interaction that the master terminal operator has with the system.

The Supervisory Terminal Operator

CICS also has facilities allowing a user to be a supervisor of a subset of the terminals within a CICS system; for example, those owned by a single department. The supervisory terminal operator has access to some privileged functions, less powerful than those available to the master terminal operator.

Monitoring and Tuning Functions

CICS Monitoring and Tuning

CICS also provides **statistics and monitoring facilities** to allow you to monitor the operation of the whole system.

You can thus tune your system and cope with changes in the patterns of demand from your users.

Statistics and Monitoring Facilities

CICS can monitor its own performance and record accounting data for individual users. Standard CICS monitoring data is collected at predefined event monitoring points (EMPs) in the CICS code. Although you can specify which classes of data to collect at these points you have no control over the actual data collected. You can also write applications to contribute your own data to user fields in certain CICS monitoring records.

There's a CICS sample program called DFHXMOLS that will list the journal data set containing your monitoring data.

The kinds of statistics available include:

- For each **transaction:** the number of times used; the amount of CPU time used; the number of times paging has been necessary; the number of times restarted after cancellation.
- For **tasks:** the peak number of concurrent tasks; the number of times the maximum task load has been reached.
- For each **terminal:** the number of input and output messages; the number of transactions; the number of transaction errors.
- For each **application program:** the number of times used.
- For each **file or DL/I data base:** the number of requests for reading, adding and updating.

Statistics collected can apply to the whole CICS system, or to selected topics in this list. You can choose the level of recording, depending on your needs for accounting, performance, or exception information. These statistics are recorded in a consistent format, giving your installation's programs a common mechanism for extracting and processing relevant information.

Analyzing the Data

CICS simply collects all these figures without analyzing them. However, you can access and print them by your own application programs, or (for certain monitoring data) by using an existing performance analysis tool such as CICSPARS (CICS/DOS/VS V1 R6 only). Analysis will help you when it comes to decisions about tuning your system to improve its performance, or when you're trying to balance resources by making permanent changes to the system.

Two of the IBM performance analysis tools available are:

- Service Level Reporter Version 2
- CICS Performance Analysis Reporting System (CICSPARS).

Service Level Reporter Version 2: This is a data reduction and analysis program which runs under OS/VS. It stores data from the system in a data base and presents it in readily understandable form. SLR can report on transactions processed, processing times, performance (on a per-transaction basis), and exception conditions that affect performance.

It can also analyze performance data collected on tapes from a VSE system.

CICS Performance Analysis Reporting System (CICSPARS): For CICS/DOS/VS users, there's a field-developed collection and reporting system called CICSPARS. This collects information on each task, including start and stop times, task number, CPU and WAIT times, and so on. CICSPARS batch programs can sort, summarize, display various online status reports and user options, and produce summary reports.

System Tuning

Any system, however carefully set up initially, is likely to change its demand patterns over time. In addition, there can be variations in the load on the system during the working day. Most systems will not need monitoring or tuning very often, but the facilities exist within CICS for when they are needed.

Figure 18 shows the relationship between processor usage and transaction response times for a reasonably well-tuned system. Since the demand pattern varies, the response time will do so also. bot For most of the time, as you can see, the response times will be either good or acceptable. Only when a given system is nearing overload will the response times start to become unacceptable. If there **is** a large variation in the system load, it will merit regular checking of system performance so that adjustments can be made if necessary. This is because the greatest scope for tuning (and consequent performance improvement) lies at the upper end of the system loading curve.

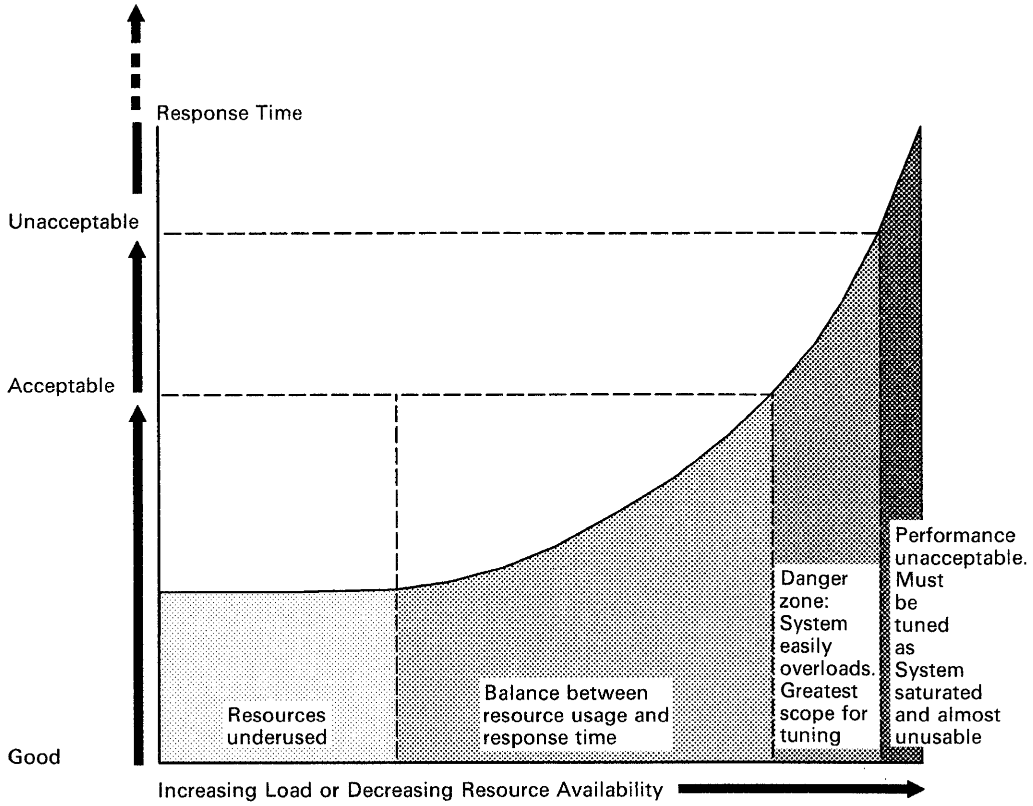


Figure 18. The Relationship Between Processor Usage and Response Time

Most performance monitoring is done by sampling information in certain CICS control blocks at regular intervals, and by counting the occurrence of critical events. The monitoring usually samples the production system in operation, for realistic results. However, monitoring uses some CPU resources and therefore affects performance. You should establish the impact that monitoring has on the performance of the system. You can then choose a volume and frequency of monitoring that gives the best balance between the performance degradation (caused by monitoring) and the performance improvement achieved by detailed control (made possible by the monitoring information).

A suitable level may well be a general monitoring of the most critical parameters and detailed investigation of each new feature or application. To help implement an efficient monitoring strategy, you should make performance testing of new applications part of the normal functional testing that each new program receives.

Please consult the *Performance Guide* for further information on CICS system performance and tuning.

Application Design Aspects

Application Design

This section describes the specific requirements of application programming for a CICS system, making suggestions about **programming methods** and **data security**.

The section ends with a look at **application programmers** themselves.

CICS Command-Level Programming

CICS application programs can be written in PL/I, COBOL, assembler language, or RPGII. (RPGII is only available on VSE systems.) Full details are contained in the *CICS/VS Application Programming Primer* (for newcomers) and the *Application Programmer's Reference Manual (Command Level)* for experienced application developers.

CICS Macro-Level Programming

The macro-level programming interface that existed before the introduction of the CICS command level interface is still supported for program compatibility and maintenance purposes. However, it does **not** support all the powerful functions introduced in recent releases of CICS. We do not recommend its use for new application programs.

Programming Methods

As you'd expect, most of the principles of good batch programming apply equally to CICS application programs. For example, all code should be well commented; GOTOs should be kept to a minimum; subroutines should be near the code that executes them, and so on. In addition, interactive application programs should be:

- **Modular**. The program should be written in small segments, each carrying out a distinct function.

  You can link a series of transactions in such a way that the end user sees the interaction as a continuous screen conversation. This enables CICS to service a large number of users through multitasking, ensuring that no single user can monopolize the system at the expense of others.

  A transaction sequence written in this way is called **pseudoconversational**.

- **Reentrant**. The program should not modify itself or store data within itself at all. This enables CICS to hold just one copy of a program in real storage even if many users are running it. CICS actually permits programs that are **quasireentrant** because such programs are easier to write than fully reentrant ones. Quasireentrant programs must not **permanently** modify themselves between their use of CICS facilities.

  The command-level interface makes reentrancy simple in both COBOL and PL/I programs, greatly increasing the efficiency of a large online system; however, RPGII programs are not reentrant.

- **Terminal oriented**. Programs must take account of the human interface with the operator. The facilities given by Basic Mapping Support for providing helpful screen layouts are likely to prove valuable. Remember, you can design maps online, in consultation with the end user department, using the Screen Definition Facility/CICS (SDF/CICS) program product.

- **CICS oriented only**. Programs must use CICS for their input, output, intertask communication, and so on. They must not try to bypass CICS by using the programming language or operating system facilities for these.

A number of sample programs are provided with CICS. They may be useful in their own right, or as sources of examples of programming techniques when your programmers are developing interactive programs for a particular installation.

Data Security

Clearly, in a system where many users have access to, and can modify, the same data, it must not be possible for unauthorized users to do so. In fact, the security requirements of any online data processing system can be summed up in the phrase: "No unauthorized access to data".

Preventing Unauthorized Access to Data

CICS provides extensive facilities under this heading. For example:

- You can require sign-on checking. For security, CICS provides an optional control table – the **sign-on table** (SNT). This holds, amongst other things, users' names, passwords, operator identification codes, and security codes. When the SNT is in use, users must first sign on to CICS with a transaction called CSSN. And if the sign-on is not accepted, CICS stores the reason for later investigation.

- For the MVS user, the Resource Access Control Facility (RACF) program product offers a variety of powerful security features that can be applied to various resources – data sets, CICS transactions, and terminals. RACF can also control access to a CICS address space from another address space.

- You can restrict access to particular application programs by giving them different security key values.

- You can use passwords or badge readers to check the identity of would-be users.

Isolating Sensitive Applications

Unless you're using MRO, CICS executes in a single partition or address space of the operating system. Thus all CICS application programs and control modules run with the same storage protection key.

As a result, it is possible (due to error or malicious intent) for one application program to affect the code of another during execution. This could matter, for example, when testing a new application program on a production system or when running a sensitive application with other applications.

You must impose thorough design procedures to minimize the risks involved. In such situations, your department may wish to run more than one independent CICS system; or to use MRO to provide isolation between untested and production programs, or between sensitive applications and others.

Application Programmers

Application programmers fall into one of two groups: designers or implementers. This does not rule out the possibility of one individual doing both tasks, of course.

Application Designers

Application designers need a good understanding of CICS general facilities. They similarly must be aware of products that CICS interfaces with, and the environments in which it runs.

They analyze the requirements of the end user departments needing the system. They convert these requirements into an application system design including the end user interface, screen formats and dialogs, recovery and restart procedures, and so on.

Application Implementers

Application implementers need a good understanding of the CICS application programming interface. Their job is to take the application designer's specifications and turn them into the required application.

They must be able to use the various CICS application development tools, the translator, and the language compiler or compilers of their choice.

Application implementers do not just implement new systems. They will from time to time be called upon to fix problems with existing systems, or to extend those systems beyond their original scope.

What's next?

In the final part of the book – the appendixes – we list some IBM program products for use with CICS, and the hardware terminals and subsystems that CICS can support.

Appendixes

The Appendixes:

- Describe some IBM software for use with CICS
- Describe the hardware that CICS can support

Opposite: Hursley House in Hampshire England, where CICS is developed

Some of the CICS development team on the lawn at Hursley House

Appendix A. Some IBM Programs for Use with CICS

This appendix may contain references to IBM programs that are not announced in your country. Such references must not be construed to mean that IBM intends to announce such programs in your country. You should consult your local IBM representative or the IBM branch office serving your locality for advice as to the availability of the programs listed below, their suitability for use with a particular release of CICS, and their current programming service classification.

The following is **not** a complete list of programs for use with CICS. For ease of reference, we've grouped the programs as follows:

- Application programs
- End user interactive products
- Data base products
- CICS aids.

Another catalog that may interest you is *Application Programs available from non-IBM sources for IBM Data Base Products*, GH20-9449. The first edition (October 1984) lists details of 112 programs, 74 of which offer CICS as an environment.

Application Programs

In this group we have:

- General library systems:
  - Storage and Information Retrieval System:
    - 5740-XR1: STAIRS/OS/VS
    - 5740-XR7: STAIRS-DL/I (OS/VS)
    - 5746-XR4: STAIRS/DOS/VS.
  - Dortmund Leuven Library System:
    - 5787-DAA: OS/VS
    - 5787-FAJ: Online OS/VS
    - 5787-DAB: DOS/VS
    - 5787-FAK: DOS.
  - Other:
    - 5796-APA: Online library circulation system (DOS)

- Text systems:
  - Document Composition Facility (DCF):
    - 5748-XX9: OS and DOS.
  - Advanced Text Management System (ATMS III):
    - 5740-XYL: ATMS III/OS/VS
    - 5746-XXU: ATMS III/DOS/VS
    - 5785-DBZ: ATMS external data interface program.
  - Integrated Processing of Data and Text:
    - 5785-DCX: Integrated processing of data and text (OS)
    - 5785-DEJ: Integrated processing of data and text, version 2 (OS)
    - 5785-DDD: Integrated processing of data and text (DOS).
    - 5785-DEK: Integrated processing of data and text, version 2 (DOS)
  - Distributed Office Support System/370 (DISOSS):
    - 5668-982: OS
    - 5666-264: DOS
    - 5798-DCW: Distributed office support for 6670 (OS)
    - 5796-BCL: Host Document View Facility.
    - 5665-290: OS version 3 release 3
    - 5666-270: DOS version 3 release 3
  - Other:
    - 5785-GBP: Online editorial system II (DOS)
    - 5796-AER: Computer generated correspondence (OS, DOS).
    - 5665-310: report management and distribution system release 2
- Banking and financial systems:
  - Interactive Financial System:
    - 5668-967: 1. Postings and general ledger (OS)
    - 5668-968: 2. P and L, balance sheet, financial status reports (OS)
    - 5668-969: 3. Open item accounting (OS)
    - 5668-970: 4. Payment processing (OS)
    - 5746-F52: 1. Postings and general ledger (DOS)
    - 5746-F53: 2. P and L, balance sheet, financial status reports (DOS)
    - 5746-F54: 3. Open item accounting (DOS)
    - 5746-F55: 4. Payment processing (DOS).
  - IBM Direct S.W.I.F.T. Link - CICS
    - 5668-964: Direct S.W.I.F.T. Network Link (DSNL)
    - 5666-269: IBM Direct Multinetwork Link (DMNL) (DOS)
    - 5668-965: IBM Direct Multinetwork Link (DMNL) (OS)
  - Automated Teller Host Support System:
    - 5798-AND: Teller control system and CICS/DOS
    - 5796-AQT: Automated teller host support system enhancement
    - 5796-ANH: CICS/DOS/VS automated teller host support system.

  - Demand Deposit Accounting and Savings Memo Post System:
    - 5796-AZF: DOS.
  - Comprehensive Thrift Application - Capital Federal Savings and Loan:
    - 5796-AXW: Host support (DOS).
  - Program customizer for the IBM 3600:
    - 5740-F11: OS
    - 5746-F11: DOS.
  - Swiss Application program for IBM 3600 FCS (SWAP):
    - 5785-LAE: OS, DOS
    - 5785-LAK: OS, DOS (SWAP/Extended).
  - Others:
    - 5666-263: Financial Management System (DOS)
    - 5665-309: Financial Management System (MVS)
    - 5798-DBA: Financial planning system - CICS / VSPC
    - 5798-DEW: Host Positive credit/check authorization (OS)
    - 5798-CNW: Check processing archive/retrieval (OS)
    - 5796-AQQ: Credit card system - authorizations (DOS)
    - 5796-ATR: Transaction cross reference 3600 4700 (OS, DOS)
    - 5748-F55: 3600 4700 administrative applications.
    - 5796-PQY: Fundamental Accounting Principles Courseware Fox Valley Technical Institution
    - 5796-PQX: GDDM File Display Maintenance Colonial Mortgage Service Company
    - 5796-BDA: IBM 3624/3614 Online Application and Reporting System
    - 5796-BDC: First American Bank of Nashville

- Public utility industries:
  - Distribution Construction Information System (DCIS)
    - 5712-AAB: Application design service.

- Communications Oriented Production Information and Control System (COPICS):
  - 5740-XYX: COPICS online routing (OS)
  - 5746-XY1: COPICS online routing (DOS)
  - 5785-GBA: COPICS bill of materials online II
  - 5785-GBC: COPICS bill of materials batch utilities
  - 5785-GBD: COPICS product cost calculations II
  - 5785-GBE: COPICS inventory accounting II
  - 5785-GAX: COPICS and purchasing
  - 5785-GBF: COPICS advanced function material requirements planning II
  - 5785-GAZ: COPICS and receiving inspection
  - 5785-EBZ: COPICS multiple system extension
  - 5785-DCN: COPICS customer order servicing - data management
  - 5785-DCP: COPICS customer order servicing - order management
  - 5798-DCQ: COPICS shop order release II
  - 5798-DCR: COPICS inventory planning and forecasting

  - 5798-CZH: COPICS facilities data control.
  - 5798-DFR: COPICS plant monitoring and control system
  - 5798-DFT: COPICS plant monitoring and control system
  - 5668-908: COPICS shop order load analysis and reporting
  - 5668-907: COPICS facilities data management
  - 5668-906: COPICS customer order servicing - shipping management
  - 5668-898: COPICS master production schedule planning

- Conversational and Interactive Project Evaluation and Control (CIPREC):

  - 5740-XC3: OS
  - 5746-XC3: DOS.

- Inventory Management (INFOREM II):

  - 5798-DDR: Base
  - 5798-DDT: Allocation
  - 5798-DET: Online processor.

- General systems:

  - Distribution Management Information System (DMIS):

    - 5785-EBA: Sales forecasting (DOS)
    - 5785-EBB: Strategy planning (DOS)
    - 5785-EBS: Replenishment control (DOS)
    - 5785-EAZ: Movements and status reporting (DOS).

  - Interactive Personnel System:

    - 5740-AM4: Interactive personnel system CICS/OS/VS
    - 5746-AM1: Interactive personnel system CICS/DOS/VS.

  - CICS/VS Data Entry Facility:

    - 5785-WAE: OS, DOS.

  - Online Plant Maintenance System:

    - 5787-NAD: Equipment maintenance (DOS)
    - 5787-NAE: Spare parts management (DOS).

- Health systems:

  - Health Care Support:

    - 5746-H14: Health Care Support/Accounts
    - 5796-ANY: HCS/DLI patient care system
    - 5796-BCF: Application execution system
    - 5796-BCE: Application preparation system
    - 5798-AYG: S/370 HCS/Order communications system
    - 5798-BDG: S/370 HCS/Revenue management system
    - 5798-CCB: S/370 HCS/DLI+registration
    - 5798-CCC: S/370 HCS/DLI+admissions.

  - Other:

    - 5796-ANA: Nurse scheduling system (DOS).

  - 5796-BBF: PCS - patient management
  - 5796-AYP: PCS - application development system/cross reference
  - 5796-PQC: PCS - nursing care plans Crouse Irving Memorial Hospital

- Customer Information File:

  - 5798-AHX: Customer information file using DOS/CICS
  - 5798-AMR: Customer information file using CICS.

- Others:

  - 5798-CFJ: Alpha search inquiry system online update (OS, DOS)
  - 5736-N14: Alpha search inquiry system (OS, DOS)
  - 5796-AFX: Online payroll and personnel system (DOS)
  - 5796-AKH: Online payroll and personnel system (OS)
  - 5796-BBE: Direct store vendor delivery control (DOS DL/I)
  - 5796-ANW: Online student registration system (DOS)
  - 5796-ANP: Online newspaper circulation system (OS)
  - 5796-ANQ: Online newspaper circulation system (DOS).
  - 5796-PQR: Education Information System/Registration San Mateo Community College
  - 5665-330: IBM personal services/370 (OS)
  - 5666-318: IBM personal services/370 (DOS)
  - 5665-347: integrated processing of data and text (OS)
  - 5666-325: integrated processing of data and text (DOS)

End User Interactive Products

In this group we have:

- Virtual Storage APL:

  - 5748-AP1: VS APL

- Graphical Data Display Manager/Presentation Graphics Feature:

  - 5748-XXH: GDDM/presentation graphics feature
  - 5798-DJZ: Visual Aids Preparation System.

- Trend Analysis:

  - 5740-XYT: Trend analysis CICS/OS/VS.

- Application Prototype Environment:

  - 5785-RAB: Application prototype environment.

- Planning Control and Decision Evaluation:

  - 5740-XX8: PLANCODE/I OS/VS
  - 5746-XX9: PLANCODE/I DOS/VS.

Data Base Products

In this group we have:

- Data Language/I:

  - 5746-XX1: Data Language/I (DL/I DOS/VS) (DOS)
  - 5798-DBH: CICS/VS ISAM - DL/I bridge (OS, DOS).

- Structured Query Language/Data System:

  - 5748-XXJ: Structured Query Language/Data System (DOS).

- Relational Data Base:

  - 5740-XYR: Database 2 (OS)
  - 5668-972: Query Management Facility (OS)

- Information Management System:

  - 5740-XX2: Information Management System Version 1 (IMS/VS) (OS).
  - 5665-332: Information Management System Version 2 (IMS/VS) (OS).

- DB/DC Data Dictionary:

  - 5740-XXF: DB/DC data dictionary OS/VS
  - 5746-XXC: DB/DC data dictionary DOS/VS.

CICS Aids

Other program products of possible interest include the following collection of aids and program development tools. Some of them have already been mentioned in the text of this book:

- Application Development Aids:

  - 5740-XYF: Screen Definition Facility/CICS OS/VS
  - 5746-XXT: Screen Definition Facility/CICS DOS/VSE
  - 5664-178: Screen Definition Facility/CICS CMS
  - 5668-824: Cross System Product/Application Development
  - 5668-825: Cross System Product/Application Execution
  - 5668-918: Cross System Product/Query
  - 5798-DEC: Development Management System/CICS/VS online debugging aid
  - 5748-XC4: Development Management System/OS, DOS (DPCX)
  - 5798-DJH: Development Management System/CICS extended access (OS, DOS)
  - 5740-XC5: DMS/CICS (OS)
  - 5746-XXV: Entry Level Interactive Application System (ELIAS/1) (DOS)
  - 5748-XXK: Entry Level Interactive Application System (ELIAS/1) (VM/CMS)
  - 5796-BAJ: Generalized CICS/VS application development architecture
  - 5796-PWJ: Generalized CICS/VS application development architecture version 2
  - 5798-CFT: CICS source program maintenance online II
  - 5798-CPJ: Graphical displays - CICS/VS (OS, DOS)

- 5785-DCQ: Online user help facility
- 5796-AHJ: CICS/VS online test/debug II
- 5798-CJX: CICS/OS/VS dump reading program.
- 5798-DJX: CICS/VS formatted dump online II

- Performance Analysis Aids:

  - 5798-DAB: CICS Performance Analysis Reporting System (CICSPARS) (DOS)
  - 5668-966: Service Level Reporter Version 2 (SLR)

- Installation and Maintenance Aids:

  - 5748-MS1: Interactive Productivity Facility (IPF) (DOS)
  - 5750-AA9: MVS/SP System IPO
  - 5750-AAA: VSE System IPO/E (DB/DC)
  - 5735-OZS: Information/Management migration utility
  - 5798-DFP: Advanced Communication Function (ACF) Network Generation Aid.

- Online Training Aids:

  - 5668-012: Interactive instructional presentation system
  - 5668-011: Interactive instructional authoring system
  - 5748-XX6: Interactive instructional system
  - 5779-EAC: MVS/JES2 operator training
  - 5779-EAA: DOS/VSE operator training
  - 5779-EAJ: OS/VS job control language training
  - 5779-EAF: Terminal user training for IBM 3278/3279/8775
  - 5779-EAG: VSE/VSAM operator training
  - 5779-EAH: VM/SP operator training
  - 5779-EAL: User Training for VSE Interactive Computing and Control Facility (ICCF)
  - 5779-EAN: User Training for Conversational Monitor System (CMS)
  - 5779-EAP: Document Composition Facility Training
  - 5779-CAZ: 3270 operator training course
  - 5779-CAA: A department reporting system II (ADRS) training
  - 5779-EAK: End user products for the information center
  - 5798-DCX: CICS/VS command level programming course
  - 5798-DGJ: CMS fundamentals II
  - 5798-DFZ: COBOL interactive debug training II
  - 5798-CKH: CICS basic training
  - 5798-CBA: DL/I basic programming course
  - 5798-CFC: DL/I advanced programming course
  - 5798-DAP: DMS/CICS/VS programming course
  - 5796-PNE: Document composition training
  - 5798-DHN: Interfacing APL based products course
  - 5796-PLF: STAIRS/VS training course
  - 5798-DFX: Using TSO effectively II
  - 5798-DAN: VSAM and access method services course
  - 5798-CWZ: 3276/3278 operator training course.

- Operations Aids:

  - 5798-DAQ: Account network management program (OS)
  - 5798-DAT: Account network management program (DOS)
  - 5798-DAW: Network error management facility
  - 5785-EAY: Local display terminal simulator

- 5798-CET: Online panels entry library system
- 5798-DGE: System error management facility/CICS/VS (OS)
- 5798-DGD: System error management facility/CICS/VS (DOS)
- 5735-XX6: NCCF terminal access facility feature
- 5785-DCW: Print spooling system for CICS/VS
- 5796-AWQ: CICS/VS online prompting facility
- 5668-007: Communication oriented message system (CORMES) (OS/VS)
- 5746-XXM: Communication oriented message system (CORMES) (DOS/VS).
- 5798-DQH: CICS/VS 3270-PC file transfer program
- 5798-DKT: VSE CICS/VS Security Manager

- Other aids:

  - 5746-AM5: VSE 3270 bisynchronous pass through (DOS)
  - 5666-979: Communications facility/Host
  - 5796-ATW: 2260 2741 compatibility using VTAM/TCAM (OS)
  - 5798-CBH: 3270 screen format utility
  - 5799-BEH: 3640 CICS/VS extensions - 4331 loops (OS, DOS)
  - 5798-DBQ: CICS/VS automated event control.
  - 5668-981: X.25 NCP packet switching interface
  - 5708-EM1: 5280-3270 Emulation Support Enhanced
  - 5798-DPG: VSAM file definition aid
  - 5666-294: Planning aid for retail information system/base (DOS)

Appendix B. IBM Terminals and Subsystems Supported

The IBM terminals and subsystems supported by CICS are shown in Figure 19. It gives an overview, showing the terminal or system type and the access method supported.

Two access method interfaces are shown. Support of SNA networks was first introduced through VTAM. It is available through ACF/VTAM and, for devices also supported by these access methods, ACF/VTAME (CICS/DOS/VS) and the Record Interface introduced in ACF/TCAM Version 2 Release 2 (CICS/OS/VS). The figure shows this support under the column headed VTAM. The second interface is shown in the BTAM column. This support is available through BTAM (CICS/OS/VS) or BTAM-ES (CICS/DOS/VS).

Bear in mind that many special features of terminals and control units are transparent to programming and are therefore readily usable even though not specifically identified. In all cases, you must use the appropriate line adapters or telecommunications control units, as specified by the access method in use.

Another interface is available in CICS/OS/VS, supporting the use of TCAM with a Message Control Program. This, the TCAM GET/PUT interface, supports data streams rather than specific terminal types. CICS/OS/VS accepts any data stream from a TCAM or ACF/TCAM supported terminal that can be edited in the message handler portion of the TCAM MCP to appear as an EBCDIC, basic SCS, or 3270 data stream.

IBM 7770 support is not available through this interface. CICS/OS/VS BMS data stream support for IBM 3600, 3767, 3770 and 3790 logical units is also available through the TCAM direct interface to NCP/VS introduced in TCAM 10.

You'll find a more detailed list, including specific units and physical attachment where appropriate, in the *CICS/OS/VS Facilities and Planning Guide*, SC33-0202.

(Users of CICS Version 1 Release 6 should refer to the edition notice, to find out the order number of the previous edition of this (*General Information*) book. That edition is still valid for CICS 1.6, and it contains the more detailed list just mentioned.)

	Access Method	
Terminal/System Type	VTAM	BTAM
1050 Data Communication System		X
2260/5 Display Station		X
2740 Communication Terminal		X
2741 Communication Terminal		X
2770 Data Communication System		X
2780 Data Transmission Terminal		X
2980 General Banking Terminal System		X
3101 Display Terminal	X	X
3230 Printer	X	X
3268 Printer	X	
3270 Information Display System	X	X
3270 Personal Computer	X	X
3270 Personal Computer/G	X	X
3270 Personal Computer/GX	X	X
3287 Printer	X	
3600 Finance Communication System	X	X
3630 Plant Communication System	X	X
3640 Plant Communication System	X	
3650 Retail Store System	X	
3660 Supermarket System		X
3680 Programmable Store System	X	
3730 Distributed Office Communication System	X	
3735 Programmable Buffered Terminal		X
3740 Data Entry System		X
3767 Communication Terminal	X	X
3770 Data Communication System	X	X
3780 Data Communications Terminal		X
3790 Communication System	X	
4300 Processors	X	X
4700 Finance Communication System	X	X
5100 Portable Computer		X
5110 Portable Computer		X
5230 Data Collection System		X
5260 Retail System		X
5280 Distributed Data System	X	X
5520 Administrative System	X	X
5550 Administrative System	X	X
5937 Rugged Terminal	X	X
6670 Information Distributor	X	X
7770 Audio Response Unit		X
8100 Information System	X	X
8775 Display Terminal	X	
8815 Scanmaster	X	
Communicating Magnetic Card Selectric Typewriter		X
Displaywriter	X	X
Office System/6		X
Personal Computer	X	X
Series/1		X
System/3		X
System/7		X
System/23		X
System/32	X	X
System/34	X	X
System/36	X	X
System/38	X	X
System/370 (also 303x, 308x, and 3090 processors)	X	X
Teletypewriter Exchange Service (TWX 33/35)	X	X
World Trade Typewriter Terminal (WTTY)	X	X

Figure 19. Overview of Terminals and Subsystems Supported by CICS

Glossary

This glossary defines special CICS terms used in the library and words used with other than their everyday meaning. It includes terms and definitions from the *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699. In some cases the definition given is not the only one applicable to the term, but gives the particular sense in which we've used it.

American National Standards Institute (ANSI) definitions are preceded by an asterisk.

A

abend. Abnormal end of task.

ACB. Access method control block. (VTAM and VSAM)

access method. A technique for moving data between main storage and input/output devices.

ACF. Advanced Communication Function.

activity keypoint. A keypoint written to the system log during normal operation of CICS. In the event of an uncontrolled shutdown and subsequent emergency restart, activity keypoints can shorten the process of backward scanning through the system log. Activity keypoints are written automatically by the system (system activity keypoints) or by the user (user activity keypoints). (See also **keypoint.**)

address space. The complete range of addresses that is available to the programmer.

addressing. In data communication, the means whereby the originator or control station selects the unit to which it is going to send a message.

ANSI. American National Standards Institute.

application. This refers to a set of one or more **application units of work** designed to fulfill a particular need (or needs) of the user organization.

application program. (1) A program written for or by a user that applies to the user's work. (2) In data communication, a program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application unit of work. A set of actions within an application that the designer chooses to regard as an entity in its own right. It is for the designer to decide how (if at all) an application should be subdivided into application units of work, and whether any application unit of work shall consist of just one or of many (CICS) **logical** units of work (LUWs).

Typically, but not exclusively, an application unit of work would correspond to a CICS **transaction.**

* **argument.** (1) An independent variable. (2) Any value of an independent variable.

* **ASCII.** American National Standard Code for Information Interchange.

assembler language. A source language that includes symbolic machine language statements in which there is a one-to-one correspondence with the instruction formats and data formats of the computer.

audit trail. A manual or computerized means for tracing the transactions affecting the contents of a record.

auxiliary storage. Data storage other than main storage; for example, storage on magnetic tape or direct access devices.

B

backout. A general term meaning to restore a previous state of all or part of a system. See dynamic transaction backout.

Basic Mapping Support (BMS). A facility which moves data streams to and from a terminal. It provides device independence and format independence for application programs.

batch. An accumulation of data to be processed.

BDAM. Basic Direct Access Method. An access method used to retrieve or update particular blocks of a data set on a direct access device.

bit map. In the context of temporary storage, a control block used by intrapartition transient data to show the VSAM control intervals (or BDAM tracks) that have been used and are available. It is updated whenever a control interval (or track) is assigned to or released from a destination.

blocking. The process of combining two or more records into one block.

BMS. See Basic Mapping Support.

BSC. Binary synchronous communication.

BTAM. Basic telecommunications access method.

byte. In System/370, a sequence of eight adjacent binary digits that are operated on as a unit.

C

cataloged procedure. A set of control statements that has been placed in a library and can be retrieved by name.

CEMT. The master terminal transaction.

CEST. The supervisory terminal transaction.

CICS. Customer Information Control System

COBOL. Common business-oriented language. An English-like programming language designed for business data processing applications.

command. In CICS, an instruction similar in format to a high-level programming language statement. (Contrast with macro.) CICS commands invariably include the verb EXECUTE (or EXEC). They may be issued by an application program to make use of CICS facilities.

command-language statement. In CICS, synonym for command.

committed change. A change that will not be backed out in the event of a failure. Changes made by an LUW are committed when the sync point at the end of the LUW is complete.

committed output message. A message that is transmitted as a result of an LUW completing a sync point (at which time changes to data resources made by the LUW will also be committed). A committed output message is one that, in the event of a failure, needs to be transmitted and acknowledged to be sure that logical consistency with the changes to data resources is maintained.

common system area (CSA). A major CICS storage control block.

communication area (COMMAREA). An area that is used to pass data between tasks that communicate with a given terminal. The area can also be used to pass data between programs within a task. The communication area is available to command-level transactions, but not to macro-level transactions.

* **concurrent.** Pertaining to the occurrence of two or more activities within a given interval of time.

control area. Synonym for control block.

control block. In CICS, a storage area used to hold dynamic data during the execution of control programs and application programs. Synonym for control area. Contrast with control table.

control table. In CICS, a storage area used to define or describe the configuration or operation of the system in a relatively permanent way. Contrast with control block.

conversational. Pertaining to a program or a system that carries on a dialog with a terminal user, alternately accepting input and then responding to the input quickly enough for the user to maintain his train of thought.

CSA. Common system area.

customize. To enhance or extend an IBM program by using services and built-in facilities provided by IBM for this purpose.

D

daisy-chain. In CICS intercommunication, the chain of sessions that results when a system requests a resource in a remote system, but the remote system discovers that the resource is in a third system and has itself to make a remote request.

DAM. Direct access method.

DASD. Direct access storage device.

* **data base.** A collection of data fundamental to a system.

* **data communication.** The transmission and reception of data.

data independence. In CICS, the ability to request data by a high-level data-management method without concern as to the mechanics of data storage or retrieval. DL/I provides application programs with greater data independence.

data integrity. The quality of data that exists as long as accidental or malicious destruction, alteration, or loss of data are prevented.

data link protocol. A set of rules for data communication over a data link in terms of a transmission code, a transmission mode, and control and recovery procedures.

data security. The protection of data against unauthorized disclosure, transfer, changes, or destruction, whether accidental or intentional.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data stream. All data transmitted through a data channel in a single read or write operation.

DB/DC. Data-base/data-communication.

DCT. Destination control table.

deadlock. (1) Unresolved contention for the use of a resource. (2) An error condition in which processing cannot continue because each of two elements of the process is waiting for an action by, or a response from, the other.

deblocking. The process of removing each logical record from a block.

destination control table. A table describing each of the transient data destinations used in the system, or in connected CICS systems.

device independence. An application program written in such a way that it does not depend on the physical characteristics of devices. BMS provides a measure of device independence.

direct access storage. (1) * A storage device in which the access time is in effect independent of the location of the data. (2) A storage device that provides direct access to data.

dispatch. To allocate time on a processor to jobs or tasks that are ready for execution.

DL/I. Data Language/I. An IBM data base management facility provided by DOS DL/I and IMS data base program products.

DOS. Disk Operating System.

DPCX. Distributed Processing Control Executive on IBM 8100 Series.

DTB. See dynamic transaction backout.

DTP. Distributed transaction processing.

dump control. The CICS element that provides storage dumps for help during testing.

dynamic log. An area in main storage used (by the journal control program) for storing copies of all changes to recoverable resources that might be required for dynamic backout of an LUW. Every execution of a transaction that has dynamic transaction backout specified has an associated dynamic log area.

dynamic transaction backout. The process of canceling changes made to stored data by a transaction following the failure of that transaction for whatever reason.

E

*** EBCDIC.** Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

EDF. Execution (command-level) diagnostic facility for testing command-level programs interactively at a terminal.

EIB. EXEC interface block.

emergency restart. The CICS facility for use following a system failure. It restores the data files of all interrupted transactions to the condition they were in when they started.

end user. In CICS, a person using a terminal to cause execution of a CICS transaction. Typically, a non-data-processing professional, for example, a reservation clerk.

enqueued. The state of a task scheduled to update a physical segment of a data base when another task is currently accessing that segment.

exception. An abnormal condition such as an I/O error encountered in processing a data set or a file.

EXEC. EXECUTE (as used in a CICS command).

F

FBA. Fixed-block-architecture. A disk storage device that stores data in blocks of fixed size. These blocks are addressed by block number relative to the beginning of the particular file.

FCT. File control table.

FERS. Facility Error Recognition System. A communications problem determination aid.

* **file.** A set of related records treated as a unit, for example, in stock control, a file could consist of a set of invoices.

file control. The CICS element that controls all CICS file operations.

file control table. A table containing the characteristics of the files accessed by file control.

* **format.** The arrangement or layout of data on a data medium. In CICS, the data medium is usually a display screen.

format independence. The ability to send data to a device without having to be concerned with the format in which the data will be displayed. The same data may appear in different formats on different devices.

function request shipping. The process that allows command-level application programs to address resources in any connected CICS region.

G

GAM/SP. Graphics Access Method/System Product

GDDM. Graphical Data Display Manager.

* **generate.** To produce a computer program by selection of subsets from skeletal code under the control of parameters.

H

* **hardware.** Physical equipment used in data processing, as opposed to computer programs, procedures, rules, and associated documentation. Contrast with software.

host processor. The primary or controlling computer in a multiple computer installation.

HPO. High Performance Option. (CICS/OS/VS only)

I

* **I/O.** Input/Output.

ICR. Independent component release.

IMS/VS. Information Management System/Virtual Storage. A general purpose system that enhances the capabilities of OS/VS for batch processing and telecommunication.

inflight task. A task which, at the time of an abnormal termination of the system, had caused records to be written to the system log, but whose processing was only part of the way through an LUW.

inquiry. A request for information from storage; for example, a request for the number of available airline seats.

installation. (1) A particular computing system, in terms of the work it does and the people who manage it, operate it, apply it to problems, service it and use the work it produces. (2) The task of making a program ready to do useful work. This task includes generating a program, initializing it, and applying PTFs to it.

integrity. See data integrity.

intent scheduling. Ensuring that a particular segment type of a data base is only accessible for potential update by one task at a time.

interactive. Pertaining to an application in which each entry calls forth a response from a system or program, as in an inquiry system or an airline reservation system. An interactive system may also be conversational, implying a continuous dialog between the user and the system.

intercommunication facilities. A term covering intersystem communication (ISC) and multiregion operation (MRO).

intersystem communication. Communication between separate systems by means of SNA facilities.

interval control. The CICS element that provides time-dependent facilities.

intrapartition destination. A queue of transient data used subsequently as input data to another task within the CICS partition or region.

IPL. Initial Program Load.

ISAM. Indexed Sequential Access Method.

ISC. Intersystem communication.

J

journal. A chronological record of the changes made to a set of data; the record may be used to reconstruct a previous version of the set.

journaling. Recording transactions against a data set in such a way that the data set can be reconstructed by applying transactions in the journal against a previous version of the data set.

K

keypoint. A set of records that describes the status of the system at a particular moment in time. Keypoint information includes extracts from system tables and control blocks such as: TCAs, FCT, PCT, DCT, TST. (See also **activity keypoint**, and **warm keypoint**)

keyword. (1) A symbol that identifies a parameter. (2) A part of a command operand that consists of a specific character string.

L

line. (1) On a terminal, one or more characters entered before a return to the first printing or display position. (2) A string of characters accepted by the system as a single block of input from a terminal, for example, all characters entered before a carriage return or all characters entered before the terminal user hits the attention key.

link pack area. (1) In MVT, an area of main storage containing reenterable routines from system libraries. Their presence in main storage saves loading time when one is needed. (2) In OS/VS2, an area of virtual storage containing reenterable routines that are loaded at IPL time and can be used concurrently by all tasks in the system.

linkage editor. A computer program used to create one load module from one or more independently-translated object modules or load modules by resolving cross references among the modules.

*** loader.** A routine, commonly a computer program, that reads data into main storage.

local. In data communication, pertaining to devices that are attached to a controlling unit by cables, rather than data links.

local device. A device, such as a terminal, whose control unit is directly attached to a computer's data channel. No data link or control unit is used. Contrast with remote device.

logging. The recording (by CICS) of recovery information onto journal 01 (the system log).

logical unit. In SNA, a port through which a user gains access to the services of a network.

logical unit of work (LUW). A sequence of processing actions (data base changes for example) that must be completed before any of the individual actions can be regarded as committed. When changes are committed (by successful completion of the LUW and recording of the sync point on the system log), they do not need to be backed out after a subsequent failure of the task or system. The **end** of an LUW is marked in a transaction by a sync point — issued either by the user program or by CICS at the end of task. In the absence of user sync points, the entire task is an LUW.

LU. See logical unit.

LUW. See **logical unit of work**.

M

macro. In CICS, an instruction similar in format to an assembler language instruction. (Contrast with command.)

main storage. Program-addressable storage from which instructions and data can be loaded directly into registers for subsequent execution or processing. See also real storage, storage, virtual storage.

map. In CICS, a format established for a page or a portion of a page.

master terminal. In CICS, the terminal at which a designated operator is signed-on.

master terminal operator. Any CICS operator authorized to use the master terminal functions.

message control program. In ACF/TCAM, a specific implementation of an access method, including I/O routines, buffering routines, activation and deactivation routines, service facilities, and SNA support.

message integrity. A facility that can be specified for a transaction that works with a VTAM logical unit. Message integrity means that every SEND performed by CICS on behalf of a transaction will flow with definite response requested. Message integrity is implied by specifying message protection.

message performance option. The improvement of ISC performance by eliminating sync point coordination between the connected systems.

message protection. A facility that involves the logging of sequence numbers (and texts) of inbound and outbound messages for selected transactions designated as **message protected**; message protection relates only to transactions that communicate with VTAM logical units.

message recovery. A general term that embraces **message protection** and **message representation.**

message representation. A facility that resends outbound messages if message resynchronization shows it to be necessary. Message representation is possible only for transactions that (1) communicate with VTAM logical units that support the Set and Test Sequence Number command, and (2) have message protection specified.

message resynchronization. A CICS function that uses the Set and Test Sequence Number (STSN) command to establish the sequence numbers of the latest inbound and outbound messages exchanged with a VTAM logical unit, and to check that those messages are in step with each other.

message switching. In a data network, the process of routing messages by receiving, storing, and forwarding complete messages.

*** modularity.** The extent to which a system is composed of modules.

MRO. Multiregion operation.

multiprogramming. * Pertaining to the concurrent execution of two or more computer programs by a computer.

multiregion operation. Communication between CICS systems in the same processor without the use of SNA network facilities.

multitasking. * Pertaining to the concurrent execution of two or more tasks by a computer.

multithreading. Pertaining to the concurrent operation of more than one path of execution within a computer. In CICS, the use, by several transactions, of a single copy of an application program.

MVS. Multiple Virtual Storage. An alternative name for OS/VS2 Release 3.

MVT. Multiprogramming with a variable number of tasks.

N

NCP. Network Control Program. A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

network. (1) An interconnected group of nodes. (2) The assembly of equipment through which connections are made between data stations.

network configuration. In SNA, the group of links, nodes, machine features, devices, and programs that make up a data processing system, a network, or a communication system.

nonswitched connection. A connection that does not have to be established by dialing.

NTO. Network Terminal Option. An IBM program product that extends the capabilities of the ACF/NCP to support a select group of non-SNA devices.

O

online. (1) * Pertaining to a user's ability to interact with a computer. (2) * Pertaining to a user's access to a computer via a terminal. The term ″online″ is also used to describe a user's access to a computer via a terminal.

operating system. Software that controls the execution of programs; an operating system may provide services such as resource allocation, scheduling, input/output control, and data management.

Operating System/Virtual Storage (OS/VS). A compatible extension of the IBM System/360 Operating System that supports hardware and the extended control facilities of System/370.

OS. Operating System.

OS/VS. Operating System/Virtual Storage.

P

*** parameter.** A variable that is given a constant value for a specified application and that may denote the application.

partition. A fixed size subdivision of main storage, allocated to a system task. Contrast with region.

password. A unique string of characters that a program, computer operator, or user must supply to meet security requirements before gaining access to data. The password is confidential, as opposed to user identification.

path length. The amount of processor execution time, for example, per task.

PCT. Program control table.

*** PL/I.** A programming language designed for use in a wide range of commercial and scientific applications.

polling. The process whereby stations are invited, one at a time, to transmit. The polling process usually involves the sequential interrogation of several data stations.

PPT. Processing program table.

pregenerated system. A CICS system distributed in a form that has already undergone the system generation process.

priority. A rank assigned to a task that determines its precedence in receiving system resources.

processing program table. A table defining all application programs valid for processing under CICS. It also keeps track of whether an application program is in main storage or not.

processor. In a computer, a functional unit that interprets and executes instructions.

program check. A condition that occurs when programming errors are detected by an I/O channel.

program control. The CICS element that manages CICS application programs.

program control table. A table defining all transactions that may be processed by the system.

program isolation. Ensuring that only one task at a time can update a particular physical segment of a DL/I data base.

programmable terminal. A user terminal that has computational capability.

pseudoconversational. CICS transactions designed to appear to the operator as a continuous conversation occurring as part of a single transaction.

PTF. Program Temporary Fix. A temporary solution or by-pass of a problem diagnosed by IBM field engineering as the result of a defect in a current unaltered release of the program.

Q

quasireentrant. Applied to a CICS application program that is serially reusable between entry and exit points because it does not modify itself or store data within itself between calls on CICS facilities.

queue. A line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a message routing system.

R

RACF. The Resource Access Control Facility program product. An external security management facility available under MVS.

real machine. An actual computer and its associated devices. Contrast with virtual machine.

real storage. The main storage in a virtual storage system. Physically, real storage and main storage are identical. Conceptually, however, real storage represents only part of the range of addresses available to the user of a virtual storage system.

recovery routine. A routine that is entered when an error occurs during the performance of an associated operation. It isolates the error, assesses the extent of the error, and attempts to correct the error and resume operation.

reentrant. The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks.

region. In MVT, a variable-size subdivision of the dynamic area that is allocated to a job step or a system task.

remote. In data communication, pertaining to devices that are connected to a data processing system through a data link.

remote device. A device, such as a terminal, connected to a data processing system through a data link. Contrast with local device.

resource. Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs.

* **response time.** The elapsed time between the end of an inquiry or demand on a data processing system and the beginning of the response. For example, the length of time between an indication of the end of an inquiry and the display of the first character of the response at a user terminal.

restart point. That point in the application dialog at which the end user recommences work after a system failure.

* **rollback.** A programmed return to a prior checkpoint. In CICS, the cancellation by an application program of the changes it has made to all recoverable resources during the current logical unit of work.

RPGII. Report Program Generator, version 2. A commercially oriented programming language specifically designed for writing application programs that meet common business data processing requirements. (RPG is available only with CICS/DOS/VS.)

S

SAM. Sequential Access Method.

screen page. The amount of data displayed, or capable of being displayed, at any one time on the screen of a terminal.

SCS. SNA character string.

SDF/CICS. Screen Definition Facility. An online application development program product used to define or edit BMS maps interactively.

SDLC. Synchronous Data Link Control. A communications protocol.

security. Prevention of access to or use of data or programs without authorization.

sequential data set. A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape.

Service Level Reporter Version 2. A data reduction and analysis program product. Useful for the analysis of CICS operating statistics.

SIT. System initialization table.

SMP. System modification program (OS/VS) used for CICS installation and service.

SNA. See Systems network architecture.

* **software.** Programs, procedures, rules, and any associated documentation pertaining to the operation of a computer system. Contrast with hardware.

SQL/DS. Structured Query Language/Data System. A relational data base management facility.

SRT. System recovery table.

starter system. In CICS, a pregenerated system providing at least one version of all system programs. A starter system is ready to use with little or no assembly.

startup jobstream. A set of job control statements used to initialize CICS.

storage. A functional unit into which data can be placed and from which it can be retrieved. See main storage, storage, virtual storage.

storage control. The CICS element that obtains working storage areas.

storage protection key. An indicator that appears in the current program status word whenever an associated task has control of the system. This indicator must match the storage keys of all main storage blocks that the task is to use.

storage violation. The overwriting of a CICS storage accounting area.

subsystem. A secondary or subordinate system.

supervisory terminal operator. Any CICS operator whose security key(s) allow use of the supervisory terminal functions.

surrogate TCTTE. In transaction routing, a TCTTE in the transaction-owning region that is used to represent the terminal that invoked or was acquired by the transaction.

SVS. Single Virtual Storage System.

switched connection. A connection that is established by dialing.

sync point. Synchronization point. An intermediate point in an application program at which updates or changes are logically complete. CICS recovery and restart facilities need not back out changes prior to a sync point. See also logical unit of work.

synchronization point. A point in the processing of a task (marked by a SYNCPOINT command, end of task, or DL/I TERM call) at which data base and file changes and the most recent output message are regarded as **committed**.

system. In CICS, an assembly of hardware and software capable of providing the facilities of CICS for a particular installation.

system activity keypoint. A keypoint written to the system log automatically while CICS is running normally. (See also **activity keypoint**.)

system generation (SYSGEN). In CICS, the process of creating a particular system tailored to the requirements of a data processing installation.

system initialization table. A table containing user-specified data that will control a system initialization process.

system log. The (only) journal data set (identification='01') that is used by CICS to log changes made to resources for the purpose of backout on emergency restart.

system program. A program providing services in general support of the running of a system.

system recovery table. A table listing the ABEND or abnormal condition codes that CICS will intercept.

systems network architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

T

task. (1) A basic unit of work to be accomplished by a computer. (2) Under CICS, the execution of a transaction for a particular user. Contrast with transaction.

task control. The CICS element that controls all CICS tasks.

task switching. Overlapping of I/O operations and processing between several tasks.

TCA. Task control area.

TCAM. Telecommunications Access Method.

TCT. Terminal control table.

Telecommunications Access Method (TCAM). A method used to transfer data between main storage and remote or local devices.

temporary storage control. The CICS element that provides temporary data storage facilities.

temporary storage table. A table describing temporary storage queues and queue prefixes for which CICS is to provide recovery.

terminal. (1) * A point in a system or communication network at which data can either enter or leave. (2) In CICS, a device, often equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel.

terminal control. The CICS element that controls all CICS terminal activity.

terminal control table. A table describing a configuration of terminals, logical units, or other CICS systems in a CICS network with which the CICS system may communicate.

terminal operator. The user of a terminal.

terminal paging. A set of commands for retrieving "pages" of an oversize output message in any order.

threading. The process whereby various transactions undergo concurrent execution.

TIOA. Terminal input/output area.

TP access method. Teleprocessing access method. For example, VTAM.

trace control. The CICS element that provides a trace facility.

transaction. A transaction may be regarded as a unit of processing (consisting of one or more application programs) started by a single request, often from a terminal. A transaction may require the starting of one or more tasks for its execution. Contrast with task.

transaction backout. The cancellation, as a result of a transaction failure, of all updates performed by a task.

transaction identification code. Synonym for transaction identifier. A group of up to four characters entered by an operator when selecting a CICS transaction.

transaction identifier. Synonymous with transaction identification code.

transaction restart. The restart of a task after a transaction backout.

transient data control. The CICS element that controls sequential data files and intrapartition data.

TST. Temporary storage table.

tuning. The process of adjusting system control variables to make the system divide its resources most efficiently for the workload.

turnaround time. The elapsed time between submission of a job and the return of the complete output.

TWA. Transaction work area.

TWX. Teletypewriter exchange terminal.

U

update. To modify a file with current information.

*** USASCII.** Deprecated term for ASCII.

use map. A control block used by temporary storage control to show space used and the space available for use in auxiliary temporary storage. It is updated whenever records are added to or purged from auxiliary temporary storage.

user activity keypoint. A keypoint written to the system log by a transaction. (See also **activity keypoint.**)

user exit. A point in an IBM-supplied program at which a user exit routine may be given control.

utilities. Informal term for utility programs.

V

virtual machine. A functional simulation of a computer and its associated devices. Contrast real machine.

virtual storage. The notional storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computing system and by the amount of auxiliary storage available and not by the number of main storage locations.

VM/370. IBM Virtual Machine Facility/370.

VS. Virtual Storage.

VSAM. Virtual Storage Access Method. An access method for direct or sequential processing of fixed- and variable-length records on direct access devices.

VSE. Virtual Storage Extended. An operating system that is an extension of DOS/VS, consisting of VSE/Advanced Functions, the minimum operating system support, and other IBM-supplied program products.

VTAM. Virtual Telecommunications Access method.

W

warm keypoint. A keypoint written to the restart data set during controlled shutdown (after all system activity has ceased). During a subsequent warm restart, information in the warm keypoint is used to reestablish system tables to the status they had at controlled shutdown. (See also **keypoint.**)

working set. (1) The set of a user's pages that must be active in order to avoid excessive paging. (2) The amount of real storage required in order to avoid a thrashing condition.

WTTY. World Trade teletypewriter.

Index

READER'S COMMENT FORM

Customer Information
Control System/Virtual Storage (CICS/VS)
General Information

Order No. GC33-0155-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Number of your latest Technical Newsletter for this publication . . .

Note: *Staples can cause problems with automated mail-sorting equipment.*
Please use pressure-sensitive or other gummed tape to seal this form.

If you want an acknowledgement, give your name and address below.

Name . . . .

Job Title . . . . Company . . . .

Address. . . . .

. . . . Zip . . . .

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

GC33-0155-2

Reader's Comment Form

Fold and tape   Please Do Not Staple   Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1H,
180 Kost Road,
Mechanicsburg, PA 17055, USA

Fold and tape   Please Do Not Staple   Fold and tape

IBM®

GC33-0155-2

Printed in the USA.