Determining network problems is an important task that can be difficult and even tedious. This capability has been enhanced for teleprocessing users and service personnel through service aids developed for their use. The service aids gather error data and provide displays that assist in analyzing system quality and in determining specific problems. This paper describes the evolution and operation of these aids.

Enhanced problem determination capability for teleprocessing

by J. B. Ford

Service aids have been developed to gather error data about the physical network, maintain it in a disk data set at the host location, and provide a variety of displays on IBM 3270 devices at any terminal location, either central or remote, for analysis of system quality and for specific problem determination. One of these service aids is DEMF (Display Exception Monitoring Facility),[1–3] which is an extended version of an earlier aid, the Facility Error Recognition System (FERS).[4] FERS was first distributed to users of CICS Version 2 and later to all users of CICS/VS (Customer Information Control System/Virtual Storage).[5] IBM provides these aids to enhance the problem determination capability of computer operators, network controllers, and systems programmers and of IBM's service personnel known as customer engineers.[6] Maximum efficiency and utilization of the local and remote teleprocessing network may thus be realized. The newer aid, DEMF, announced in December 1977, extends this capability to IMS (Information Management System) and to TCAM (Telecommunications Access Method) users.

Need for network monitoring

Availability of on-line equipment in a complex system, particularly from the end user's point of view, is difficult to measure and/or control. DEMF and FERS address this problem by providing real-time displays of terminal and data link errors. This approach is a logical and physical extension of LOGREC[7] (error recording function for hardware errors): logical in that it displays error counts by line, by control unit, or by terminal which can show patterns of "hits" on the network not immediately visible in a LOGREC printout; physical in that the users or customer engineers at any remote site have instant access to this error data base and to all of the functions of the service aids. The information provided enables the central site to detect terminal and data link errors and to take corrective action, such as temporarily placing a unit out of service to prevent degradation of the system, until the unit is repaired.

At the affected location, this data (which may pertain to a failing display device, audio response unit, communications terminal, etc.) can be displayed on any available IBM 3270 display unit. If the failing unit is itself a display device, additional prompting displays are available to the customer engineer to help him interpret the sense and status information, understand its meaning, and expedite the repair.

Potential benefits of network monitoring

The two primary potential benefits of an orderly, systematic network control procedure are:

1. Increased system availability by detecting network problems quickly and taking corrective action.
2. Avoiding delay in locating other system problems if the network is *not* at fault.

When network problems do occur, the more complex the network, the less obvious it is to the central site that a problem even exists. For example, a CICS system can appear to the computer operator to be executing, yet it can be totally or partially stalled for several reasons, one of which may be hardware errors on one of the data links. No operations manager wants end users calling in to report response time variances of a few seconds—such calls are quickly discouraged—and the end users, after a certain amount of experience, will gravitate to a natural "wait-and-see-if-it-comes-back-up" state for as much as 10 or 15 minutes.

This period of unavailability is unnecessary if a network controller is able to detect a component beginning to malfunction. The performance of an on-line system is subject to many variables other than the teleprocessing equipment, for instance, disk and tape errors, program problems, paging rates, data set placement and disk queuing, or CPU utilization. Yet, the impact of degraded performance will always be felt first by the end users, and their complaints to the central site of intermittent or erratic response times will frequently cause operations to suspect network problems.

If an experienced network controller can query the teleprocessing error rates and deduce, relative to the normal error rate of an installation, that there is *no* significant network problem, then the controller can be attentive to other potential causes of the problem, such as any program changes that were made recently, data sets that have been reallocated, or a stalled task in the system that should be purged.

Hence, an aid like DEMF or FERS will be only one of many "windows" into the system that the network controller should have. The ideal teleprocessing installation should have a thorough change-control procedure and several performance aids in addition to DEMF or FERS. With these aids, an experienced, dedicated network controller can materially increase the availability seen by the end users.

History of the service aids

An early use of software recording of hardware errors was on one of the original SABRE systems.[8] Disk errors were logged to tape and the customer engineers reviewed a daily listing of those errors to guide them in servicing the disk files. Teleprocessing errors, however, were not handled in such a sophisticated manner. All terminals provided hard copy, including those in the computer room for monitoring and control (the equivalent of alternate consoles today). Every teleprocessing error anywhere on the network generated a message to one of these computer room terminals to which other messages were also routed to produce a log of all important system events but not in any orderly format, nor in machine-readable form. If a data link problem were suspected, the person solving the problem would visually scan the printed terminal output to pick out the pertinent teleprocessing errors and mentally try to establish the pattern or scope of the problem.

As the early operating systems grew into more complex ones such as IBM's OS and DOS, this concept of error recording expanded into the LOGREC function. Eventually all errors were being logged, including teleprocessing errors; however, BTAM (Basic Telecommunications Access Method)[9] not only recorded teleprocessing errors in LOGREC but also printed a console message. In some early CICS installations, the large number of terminals coming into use caused BTAM messages to tie up the console and impede other system messages.

As a solution, some users chose to modify BTAM to suppress these console messages and depend on the LOGREC printout for service on the teleprocessing network. That, however, was not a permanent solution because in a CICS installation of 40 or 50 terminals the average LOGREC file could fill up with time-out messages before the operational day was over and much sooner in some cases. The LOGREC philosophy is to record sequentially and to cease operation if the allocated space fills up; thus, a bad data link could exhaust LOGREC for the rest of the operational day.

development of FERS

The first event that led eventually to FERS developed from a request by a customer engineer maintaining a network of 96 local IBM 2260 Display Stations at the Atlanta Gas Light Company. The cables for these units exited from the control units, went under the floor, and through walls and ceilings to an information center. Although each cable and terminal was marked with identifying labels or tags, the information often quickly became unreadable, unreliable, or was missing. Service then became difficult. The customer engineer requested and was provided with an inquiry that would respond to a display unit and display that terminal's hardware address.

In a later event the author assisted a new CICS user, who had 2260 display units, in solving some application problems. This user initially had had numerous teleprocessing errors and had suppressed the BTAM error messages by modifying BTAM. A proposal was made to add code to CICS to log the errors onto disk and to program some displays of that error data, but the network soon settled down so that it became unnecessary. This proposal was beneficial in the long run, however, because it enabled the disk organization and display logic to be specified that would eventually become FERS.

In performing a conversion from IBM 2260 to 3270 display units for another user, the first cluster of new terminals would not operate properly. After trying several corrective procedures, a decision was made to look at the remote status messages. A "hook" was put into the CICS Terminal Error Program to log this information onto disk, and a simple module was coded to display this data dynamically.

Thus, the older display unit was used to look at the incoming error data of the new display units. With this ability, a transaction could be tested for such items as the residual count, the failing command, or the response characters. For each transaction, knowing the exact results of each transmission assisted in problem determination.

Further use of the program at the installation led to its enhancement, thus providing more and more useful problem determination and isolation information. Eventually, additional enhancements were added to the program; it was documented and packaged for distribution as an OS version and a DOS version.

Systems overview

FERS consists of a direct access data set and two functional programs called the *logger* and the *display* (the display function is physically several modules of code).[1] The data set consists of one physical block (any size the user wishes to allocate up to full track size) for each teleprocessing line in the CICS network. The file is initialized to zeros by an operating system utility so that the display modules will not encounter meaningless data. (The display function can also access an optional, user-supplied information file that contains static data of any nature, such as supervisor names, telephone numbers, etc.)

FERS is totally dependent on the CICS environment in which it operates. When a BTAM event is posted "complete with error," the Terminal Control Program performs some analysis and may or may not retry the event. If it decides that the I/O operation was unrecoverable, it constructs a control block called the terminal abnormal condition line entry (TACLE) for the user, gathering bits of error data from several sources in main storage. This TACLE is passed to the user-written Terminal Error Program so that the CICS system programmer may have the opportunity to take some constructive action. FERS takes advantage of this CICS-supplied interface and the commonality of the error data (the TACLE is the same for OS and DOS and for all BTAM-supported devices) and logs the TACLE onto disk tagged with the failing terminal's identification.

The placement of this data in the data set requires coordination between the logger and the display so that retrievals can be accomplished. Each physical disk block represents a physical teleprocessing line, and the CICS terminal control table (TCT) is used as a map of the physical network. Thus, if an error occurs on a terminal on the third relative line in the TCT, the logger will add that TACLE to the third relative block of the data set. This procedure allows the logger to perform its function with minimum overhead (one physical update), yet the file does not have to be searched sequentially for most of the displays.

query servicing

A query for error information about a specific terminal is serviced by locating that terminal identification in the TCT, computing its relative line number, and reading that relative block from the file. Queries often call for a recap, or listing, of errors on certain lines or devices. Except for a request for a recap of all lines, a display requires only one random access read. (Entering the request "ALL LINES" makes a display appear that lists every teleprocessing line in the system.) Table 1 lists some of the displays that can be called up.

Table 1 Sample of displays

Display	*Content*	*Purpose*
ALL LINES	Recap of errors by leased line or line group.	If system performance is suspect, to find the line causing the problem or to confirm that teleprocessing errors are not the problem.
SPECIFIC LINE	Recap of errors by control unit (or stand-alone unit) on the requested line.	If the "ALL LINES" display indicated a line in trouble, to see which control unit, if any, is having the majority of problems.
CONTROL UNIT	Recap of errors by terminal on the requested control unit, and a count of control unit errors not related to any terminal.	If the line display indicated a control unit having trouble, to determine if the control unit is failing or if a particular terminal is the problem.
SPECIFIC TERMINAL	Recap of errors on that terminal in chronological order.	Having isolated the problem terminal (and probably put it out of service), to diagnose the failure.
SPECIFIC ERROR	More-detailed data.	To diagnose probable cause of failure in greater depth.
BITS PROMPTER	Prose explanation of sense and status fields.	To prompt users who are not familiar with sense and status byte combinations.

Random access was chosen for the file design instead of a sequential organization for efficiency and performance. Once generated into the CICS system, FERS becomes an integral part of the user's production system and may be used by the network controller or by customer engineers at any time, even during peak hours of production. To gain widespread user acceptance, a service aid like this one must perform a useful service and do it efficiently with no visible impact on production or on response times.

The block size of the file may be any size the user wants to allocate, and when the logger makes its first entry into an empty block, it dynamically computes the number of TACLEs that the block can hold. This allows the file for FERS to reside on any direct access device that CICS supports without being device-dependent. As errors occur on a line, its TACLEs are logged sequentially in its relative block until that block is full. When an error arrives and must be logged in a full block, the logger "wraps" to the top of the block and begins to reuse the older slots. Hence, FERS solves the problem of running out of space and losing current information, offset by the fact that it overlays older information in order to show the network controller or the customer engineer what is happening *at the moment.*

Because of this procedure, the file is not guaranteed to be complete at all times; therefore, no hard-copy reports are produced by FERS. LOGREC continues to be the chronological history of errors on the network for analysis, planning, and predictive purposes, whereas FERS is intended to be a minute-to-minute decision-making tool.

In addition to the dynamically constructed recaps of the TACLEs, a large part of the display function is "prompting" code (in the form of preprogrammed text), that is, bit-by-bit prose explanations of sense, channel status, remote status, and the like. These secondary displays are invoked only on demand because most sense and status information is highly repetitive. The basic displays present this data in hexadecimal form, but every BTAM sense and status bit can be individually explained in technical English, and many of the more common 3270 display unit *combinations* of status bits can be queried without having to refer to a manual.

Problem determination by the user

Efficient network control requires continuous monitoring and prompt action when problems are suspect. Large systems often have additional equipment to assist with this function, but software aids like DEMF and FERS will always be important and, in many installations, will be the only problem determination aids on hand.

A network controller must be competent and knowledgeable about data processing in general, various elements of teleprocessing, and all of the user's applications. Yet a person performing this function on a full-time basis usually occupies most of the operational day monitoring a system that is having no trouble.

monitoring methods

If this function is assigned to someone with other duties, such as a computer operator or a tape librarian, then a system stall may go unnoticed for many precious minutes. Some users have resorted to such devices as sending a time-initiated display to a terminal near the console that will "flicker" or "ripple" as long as the system performs satisfactorily; thus, a part-time controller can occasionally look across the room for visual evidence on network operations.

A theoretical solution to an undetected stall conceived at Southern Company Services is the possibility of building an inverted delay circuit attached to the programmable buzzer of a display station. Working in conjunction with a user-written CICS transaction, the circuit would remain in a quiet state as long as it received a pulse from CICS every few seconds, but if the system began to deteriorate, the transaction would no longer signal the display station. The circuit would decay, close a relay to the power supply, and activate a bell or buzzer calling the network controller to immediate action. This procedure would ensure prompt attention to any condition that the user wished to program into the CICS transaction without forcing someone to monitor the system continuously.

service aid use

Whatever regimen the user works out for his installation, be it periodic inquiries or the occurrence of some event such as loss of flicker on a display, posting of a master terminal message, or ringing of a warning bell, the network controller will invoke a service aid inquiry for a recap of all lines. The response will be display of the current count of total errors on each line since the last time the file was cleared (normally the start of each production day).

If all lines show a nominal error rate, that is, the norm to be established by each user, no further action need be taken until the next periodic check unless, based on other symptoms, the network controller decides to use other aids to look for problems such as hung tasks, thrashing, or any other nonteleprocessing problem.

When a line shows substantial errors above the norm, the next inquiry would be a recap of that physical line. This display will show a count of total errors on each control unit on that line. Stand-alone devices with a built-in control unit will be displayed as if they were a cluster of one terminal.

Should all control units on that line show an even spread of errors, more or less, the network controller would have the line physically tested and/or examine the host site modem for that line. If, however, one control unit has a high error rate and all others are relatively clear, the next logical step is to call for a recap of that particular control unit.

This display will show error counts by terminal on that control unit and a count of errors that are not terminal-related but could only be caused by the control unit or data link. These counts may then lead the network controller to suspect a nonterminal problem at that location, and the controller would take steps to have that site check its modem, see that the control unit has power, or possibly arrange for a test of that leg of the line.

If the control unit display shows a specific terminal with an unusual number of errors, the next request would be for a recap of that terminal. This display lists in chronological order (as much as will fit on the screen, with commands for paging if needed) portions of each error record recorded for that terminal. The network controller will not be interested in going deeper into the meaning of the sense and status information at this point—he, or she, should, by now, have established whether or not there is a problem, if so, its probable location, and the vendor to call for service. The network controller would then turn to the immediate needs of the system: should this line, control unit, or terminal remain in service or is it consuming too much of the resources of the system? Should the unit be put out of service for the benefit of the rest of the users? Is there any alternate plan to implement, such as switched network backup?

Sometimes this decision must be made in an arbitrary and heavy-handed manner. For instance, if three clusters of 3277 display stations are multidropped on one line and the modem drifts on one of the control units, perhaps that unit does not operate properly on nine out of ten polls. All users on the line will be upset with the performance of their terminals, but the users at the failing site may object to being placed completely out of service. Such action could result in overtime work or require reversion to a tedious manual procedure. Yet, the iterations these time-outs can cause will saturate the line and seriously impede the ability of other locations to perform useful work. With impartial data provided by the service aid, the network controller can make an intelligent economic decision and optimize availability for those users on that line who can function efficiently until full service can be restored.

Problem determination by a customer engineer

To use DEMF or FERS for problem determination, the customer engineer must have access to an IBM 3277 display station or communicate with someone who is at one. Since the customer engineer is called to service a specific unit, a recap of all lines is not necessary, but a recap of the affected line and control unit might be done to be sure that the problem is localized.

In many cases the fault is visibly obvious, such as double cursors on the screen, and the customer engineer will not need status information (other than to confirm the diagnosis). If the fault is not immediately visible, the next level of attack is to perform whatever routine diagnosis is customary for that unit—often more efficient than an esoteric analysis of sense and status data if it reveals the problem. When the problem can be solved by one of these methods, the service aid may then be used to confirm that the repair has been accomplished correctly, and the unit has ceased to cause errors.

If the device in question is a modem or a terminal not of the 3270 type, the customer engineer has available to him the equivalent of all of the BTAM messages that the central site would have seen at the CPU console or CICS master terminal. If the device is a 3270 type, the additional capability to retrieve paragraphs of technical information explaining several dozen of the more common IBM 3271 and 3272 control unit sense and status bit *combinations* is available.

checking for errors

The customer engineer would begin with a recap of the specific terminal and see a list of the salient fields of the errors recorded for this unit, including sense and status. The time of each error may be meaningful because the back-to-back pattern or the intermittent nature of the times may give a clue about the problem. Or the teleprocessing operation code sequence may reveal something useful, as a continuous failure on one type of operation might be seen, such as polling, or an alternating pattern—failure on addressing always followed by failure on response to addressing—might be seen. Such patterns may reveal whether the problem exists in transmission in one direction only or in both directions, which in turn, may direct the customer engineer to a particular circuit or card.

If more information is needed, the customer engineer can move the cursor to any one of the error displays, depress ENTER, and see a complete blowup of that error. This information would include the actual response bytes to the last operation, the status flags, the event completion code, the BTAM operation code, the channel command, the original data length, and the residual count.

Each error can be examined in this manner, but after two or three the data is usually redundant, especially if it is a hard and fast repeating error. If the device being serviced is a 3270 type, the customer engineer can call for text concerning the sense and status "combinations." Each individual bit has been explained on a previous display; for example, the blowup of an error may show the "DATA CHECK" and "UNIT SPECIFY" bits on in the status field, but the question arises—is there any special significance to this combination? If DEMF or FERS is queried for the meaning of these bits on and others off, this message is received:

DURING POLLING, SELECT, OR READ COMMANDS THE ADDRESSED DEVICE DETECTS A 'CURSOR' OR 'PARITY' CHECK ON DATA THAT IS BEING TRANSFERRED TO THE CONTROL UNIT OR DURING THE COMMAND OPERATION ON A WRITE COMMAND.
DEVICE RESPONSE = RVI-TO A SELECT.
EOT-TO READ, WRITE OR POLLING COMMANDS.

To customer engineers, this message will be a meaningful explanation of an error, and this kind of detail helps solve the problem. This information is available from other sources and can be found in manuals if necessary, but FERS reduces the time such research would take and eliminates the need to carry around this information.

Evolution of FERS to DEMF

The success obtained in enhancing the problem determination capability of users and in increasing the availability of display equipment installed under CICS led to an extension in the capability to a larger set of users. In planning to adapt FERS to an IMS (Information Management System) environment, three problems were initially encountered:

1. IMS message services cannot accept the native 3270 data streams produced by FERS displays.
2. IMS has no equivalent of the CICS user-written Terminal Error Program, hence, no way to create the error data set.
3. IMS places a terminal out of service when an error occurs, thus reducing drastically the quantity of error data available.

The inability to create a useful file of error data impeded progress for several months. However, it was felt that BTAM I/O errors could be intercepted outside of CICS. With a colleague providing assistance with detailed information on the operating system internal workings, the author prepared a test system and demonstrated that BTAM errors could be intercepted at the access method level and used to create a data set independent of any teleprocessing subsystem executing in the system.

Another demonstration was later set up that included some theoretical VTAM (Virtual Telecommunications Access Method) displays of artificially created error data. Feedback from this demonstration resulted in several new ideas concerning the useability of service aids, for example, making the input easier for nontypists or making the displays clearer in their messages. A question commonly asked was: Could the same principle be applied to the servicing of nonteleprocessing equipment; that is, could an error file be queried for a disk or tape unit without having to ask for a LOGREC run? Although it could be done, resources were concentrated in the teleprocessing area at that time.

Another common request was for FERS to identify line failure or modem problems specifically and recommend to the network controller a precise service need and vendor, that is, to pinpoint a problem outside of the terminal cluster. Unfortunately, only the 3270-like displays and the terminal equipment present any sense and/or status data to the host. The rest of the data link (the host modem, the common carrier, and the remote modem or any multiplexors involved) do not inject any useful information about themselves into the data exchange.

This situation is not likely to change quickly, although we see in the new 3274/3276 equipment the ability to invoke local diagnostics of a modem. Therefore, in the short run, software aids like

DEMF are apt to be limited to the error data currently transmitted, but future service aids may find equipment other than just terminals providing error data, especially as we progress into satellite communications.

DEMF as follow-on

In developing DEMF as the follow-on to FERS, the primary objective was to create a file design that was not dependent on the CICS terminal control table so the function could be extended to non-CICS users. The error logging is invoked by an extension to the LOGREC supervisor call module rather than the CICS Terminal Error Program. With this enhanced version, the condition of a CICS line can be queried at a TCAM terminal, or a TCAM controller can be queried at an IMS terminal. In other words, if any suitable display device is available, the entire BTAM and/or TCAM network can be examined regardless of the allocation of teleprocessing devices to regions or partitions in the CPU.

DEMF further enhances the file structure by allowing the user to allocate as many disk blocks as it is possible to commit to this function; then DEMF acquires a block from this pool as needed. When a block fills up, DEMF chains another block to the filled one in a push-down manner. This procedure represents some additional overhead; if a block could contain 100 entries, an extra I/O operation would be performed on every hundredth error on that line to chain the blocks, but no data would be lost if enough space is allocated for a typical day's production. As a safety measure, however, DEMF will reuse the oldest block in the longest chain if the pool should become exhausted. Both FERS and DEMF are designed around the philosophy that it is better to work within the limits of a finite file size rather than risk bringing down the user's system with a space allocation problem.

The basic practice of logging all errors on a line into the same block or chain of blocks remains the same. This practice allows the display functions to provide various recaps that may reveal a pattern to the network controller without having to do an inordinate amount of I/O operations. If errors were indexed by terminal identification, the entire file would have to be read to recap all hits on a line or control unit. But the direct access organization makes it possible to program displays that read the appropriate block for that line and sort in main storage those errors that pertain to the specific request.

Concluding remarks

The VTAM era is new, but the trend is clear. The end users want more availability, not more information about why the system is unavailable, but central host control will become more complicated. With distributed intelligence, alternate CPU paths, and intelligent controllers that can "coast" when the link to the host is lost, *apparent* availability may increase without any corresponding improvement in hardware meantime-to-failure rates, but such super systems are going to require more sophisticated error analysis at the control program level in an integrated, cohesive manner, not added on as was FERS.

Microcoded service aids for customer engineers already exist in some of the new SDLC (synchronous data link control) devices; for example, the IBM 3601 controller records incidents on its minidisk, the IBM 3767 communication terminal records in microcode storage. Additional microcode allows the customer engineer to review this data. With the sense and status information currently available, it would be possible to extend the analytic capability of a network aid like DEMF or FERS by doing statistical analysis and probability computations that could create predictive reports. A daily or weekly trend analysis of hardware problems could be very useful to the network controller.

Development of aids such as FERS may not be as urgent as in the BTAM era if the objectives of distributed intelligence are successfully met. Yet, the network controller may be faced with more important problems such as the integrity or security of the distributed data bases at the remote locations. Knowledge that something unusual has happened on a loop or at a node may be even more critical than today's need to know that a terminal or control unit is not operating properly.

As the environment changes, the nature of the information needed at the host site will change, and the relative simplicity of the original approach will not continue to suffice. New vistas are opening up with the announcement of ACF (Advanced Communications Function) and NOSP (Network Operation Support Program). For example, one can envision programming an automatic switchover to a "warm" backup system during a catastrophe without operator intervention. With the announcement of DEMF as a Selectable Unit, we see a growing number of aids becoming available to assist in network control.

ACKNOWLEDGMENTS

The theoretical solution to an undetected stall was conceived by Harry V. Early, Manager of System Support at Southern Company Services. William A. Mashburn, Jr. of IBM's Field Engineering Division provided assistance with the internal workings of the operating system in the test system used to demonstrate the interception of BTAM errors.

CITED REFERENCES AND FOOTNOTES

1. *OS/VS Display Exception Monitoring Facility (DEMF) System Control Programming Specifications*, GC34-2001, IBM Corporation, Data Processing Division, White Plains, New York 10604.
2. *OS/VS Display Exception Monitoring Facility (DEMF) System Information*, GC34-2002, IBM Corporation, Data Processing Division, White Plains, New York 10604.
3. *OS/VS Display Exception Monitoring Facility (DEMF) User's Guide*, GC34-2003, IBM Corporation, Data Processing Division, White Plains, New York 10604.
4. *Facility Error Recognition System (FERS) Service Aid Description*, GC29-7031, IBM Corporation, Data Processing Division, White Plains, New York 10604.
5. *Customer Information Control System/Virtual Storage (CICS/VS) System Programmer's Reference Manual*, SH20-9004, IBM Corporation, Data Processing Division, White Plains, New York 10604.
6. FERS and DEMF are distributed to users upon request by an IBM Programming Support Representative.
7. *OS/VS SYS1.LOGREC Error Recording*, GC28-0668, IBM Corporation, Data Processing Division, White Plains, New York 10604.
8. SABRE, the name of the earliest passenger name reservation system for an airline, was derived from the acronym SABER which, in turn, came from Semi-Automatic Business Environment Research.
9. *Basic Telecommunications Access Method*, GC30-2004, IBM Corporation, Data Processing Division, White Plains, New York 10604.

GENERAL REFERENCE

1. *Program Information, System/370*, GB21-9949, IBM Corporation, Data Processing Division, White Plains, New York 10604.

Reprint Order No. G321-5075.