Aspects of IMS/VS communication facilities discussed include networks, terminals, security facilities, editing, and formatting. Other parts in this series on IMS/VS include an introductory part on objectives and architecture, data base facilities, batch processing facilities, and transaction processing.

The information management system IMS/VS Part IV: Data communication facilities

by W. C. McGee

The Data Communication (DC) feature of IMS/VS provides facilities that permit users to communicate through terminals with a single on-line execution of the system. Communication may take any of the following three forms: (1) transmission of messages between terminals and user-written application programs; (2) transmission of messages between terminals; and (3) transmission of commands and command responses between terminals and the on-line execution.

Multiple Systems Coupling (MSC) extends the capabilities of Data Communication by providing for communication between two or more on-line executions that are running in the same or different CPUs. Messages may be transmitted between the terminals or programs of one on-line execution and the terminals and programs of a second execution.

The Data Communication facilities of IMS/VS are described here by first discussing facilities for defining physical and logical networks. This is followed by a description of logical network operation. Finally, a discussion is given of the security and editing facilities of the system, and facilities for controlling network operation.

Physical networks

The basic network configuration of IMS/VS with the DC feature consists of a single on-line execution running in a host CPU, and a set of communication controllers, lines, and terminals that are associated with the on-line execution. A variety of terminal types is supported, as indicated by the following list:

- Printer-keyboard (typewriter) terminals, e.g., the IBM 2740.
- Display-keyboard terminals, e.g., the IBM 3277 display station.
- Special-purpose input/output devices, e.g., the magnetic card reader on the IBM 3277 display station.
- Printers, e.g., the IBM 3286 printer associated with the IBM 3270 information display system.
- Batch data entry devices, e.g., the IBM 2770 data communication system.
- Intelligent controllers, e.g., the IBM System 7 when used as a controller.
- Data transmission terminals, e.g., the IBM 2780 data transmission terminal.

Terminals in remote locations are connected to the computer via communication lines. Both *nonswitched* and *switched* lines are supported. With a nonswitched line, a permanent path is maintained between the computer and one (*single-drop*) or many (*multidrop*) terminals. With a switched line, a path is established as required through common carrier switching equipment. Terminals on nonswitched and switched lines are called *nonswitched terminals* and *switched terminals*, respectively.

Certain terminal types located near the computer may be connected directly to the computer via its multiplexer or block multiplexer channels. These are called *local terminals*. Examples are the IBM 3277 display station and the IBM 3286 printer. The card readers and printers used as system input and output devices may also be defined to IMS/VS as local terminals.

The Automobile Club of Michigan currently uses approximately two hundred 3277 display stations and one hundred IBM 3284 printers for communication with the central computer. A typical branch office has two or three display stations and one or two printers connected to a single IBM 3271 control unit. Control units are multidropped from nonswitched lines, with an average of between four and five control units on a line. Twelve lines are used, of which two are 4800-baud lines and the remainder are 2400-baud lines. The longest line, connecting Dearborn and Traverse City, is about three hundred miles long. In addition to the remote terminals, the network includes a number of locally attached 3277s and 3284s.

The MSC feature permits the pairwise interconnection of up to 255 basic DC feature configurations. Two on-line executions running in the same CPU may be interconnected via main storage or I/O channels. Two executions in different CPUs may be interconnected via I/O channels or Binary Synchronous Communication (BSC) lines.

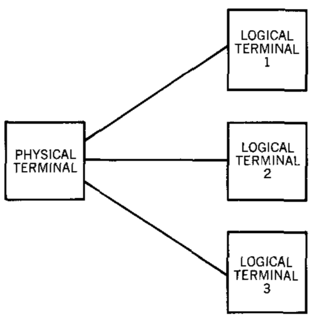
The parameters of each basic network configuration are declared in a separate system definition. When the Basic Telecommunication Access Method (BTAM) is used, a fairly extensive definition is required in order to provide a high-level communications interface to application programs. Entities defined include line groups (lines with terminals that have common attributes), lines, terminals, and terminal components. With the Virtual Telecommunication Access Method (VTAM), a simpler IMS/VS definition is possible because VTAM itself requires a more complete definition. Under VTAM, only terminals (VTAM logical units) are declared to IMS/VS, with logical unit names replacing terminal addresses.

When two configurations are connected by an MSC link, a definition of the link must be included in the definition of both configurations. Physical link parameters include name, type, and address of the channel or line that implements the link.

Logical networks

Because of the large number of terminal types supported by IMS/VS, it has been found useful to provide the concept of a *logical terminal.* A logical terminal is an abstraction of a real terminal, in which only those aspects necessary for the writing of application programs and operation of the system are apparent. All other details are suppressed. A logical input terminal is capable of generating character strings; a logical output terminal is capable of accepting character strings; and a logical input/output terminal is capable of doing both. Logical terminals are assigned one-to-eight character names through which they are addressed.

Figure 1 Physical-logical terminal relationship



Logical terminals are defined in terms of physical terminals. Any number of logical terminals may be associated with a given physical terminal, as illustrated in Figure 1. The logical terminals are in sequence, and any input generated by the physical terminal is considered to come from the first logical terminal in the set that has been enabled by system definition or master terminal command. Output directed to a logical terminal is sent to the associated physical terminal. Each logical terminal may have different sets of attributes, and by enabling and addressing the appropriate logical terminal, these attributes may be imparted to the underlying physical terminal.

Logical terminal assignments for nonswitched terminals are made during system definition. For example, the statements

LINEGRP DDNAME=DD3277R1, UNITYPE=3270 . . .

LINE ADDR=090

TERMINAL ADDR=4040, UNIT=3275, FEAT=(PFK,NOCD,NOPEN)

NAME KA55110

NAME KA55111

associate the logical terminals KA55110 and KA55111 with the 3275 display unit at address 4040 on line 090. Assignments may be changed by master terminal command during operation. For example, logical terminal KA55110 may be reassigned to the second physical terminal on line 2 with the following command:

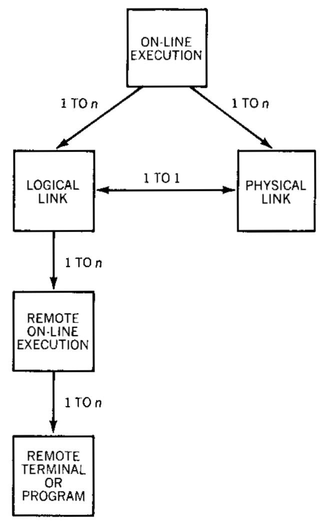
/ASSIGN LTERM KA55110 TO LINE 2 PTERM 2

Thus, if a physical terminal becomes inoperative, the logical terminals associated with it may be reassigned to another physical terminal and programs that address these logical terminals will continue to operate.

For switched terminals, logical terminal assignments are made when the physical terminal is connected to the computer and the operator issues a sign-on command. A pool of logical terminals is maintained by IMS/VS for each switched line or line group, and the terminal operator selects one or more logical terminals from this pool to be associated with the physical terminal for the current sign-on. The pooling of logical terminals makes it unnecessary to dedicate a set of logical terminals to each of a possibly large number of switched physical terminals.

For purposes of controlling system operation, the user must designate one logical terminal as the *master terminal.* The master terminal is distinguished from other terminals (user terminals) in that it is the only terminal through which certain reserved commands can be entered, and the only terminal to receive certain system messages.

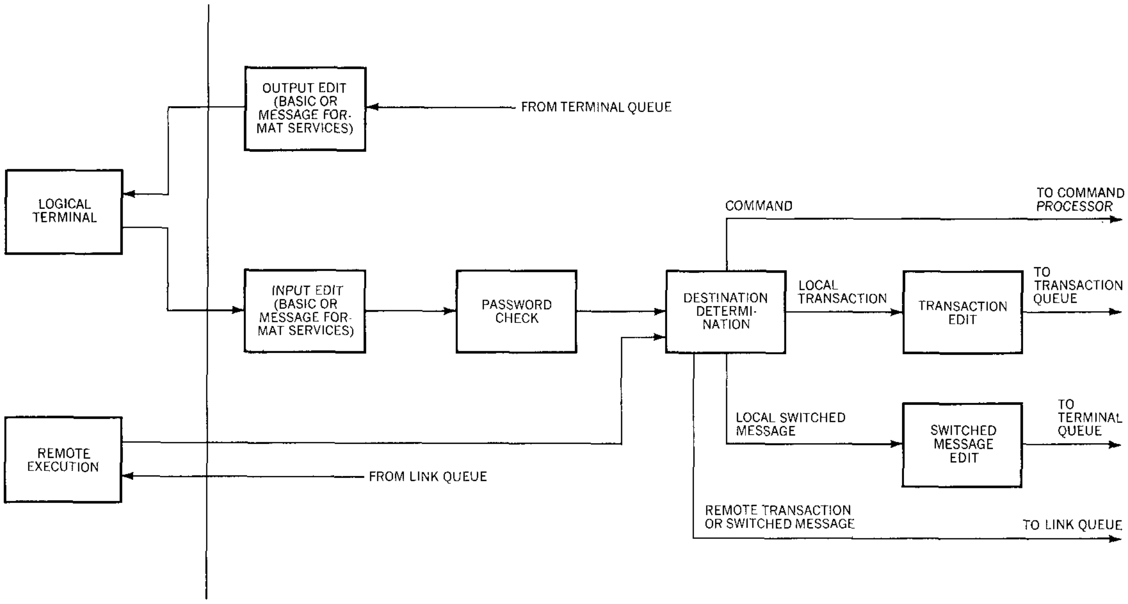
Figure 2 Multiple System Coupling associations in a single on-line execution



By analogy with the logical terminal concept, the MSC feature provides the concept of a *logical link.* A logical link is an abstract connection between two on-line executions. It is the entity to which one execution directs messages for transmission to another execution. Before the link can be used, it must be associated on a one-for-one basis with a physical link that connects the two executions, as shown in Figure 2. Once this association is made, messages directed to the logical link are transmitted over the associated physical link.

The definition of a logical link and its association with a physical link appears in the definitions of both executions that are connected by the link. The physical link association may be altered during operation by master terminal command (from both

Figure 3 Communication input and output



executions), thus permitting the logical link to be reassigned to another physical link in case the one it is currently assigned to becomes unavailable.

A logical link is also associated through system definition with one or more remote logical terminals and application programs, i.e., terminals and programs that are controlled by some other execution. Messages that have remote destinations are directed to the associated logical link. By defining such associations in each execution of the network, a message may be routed through any number of intermediate executions. These associations may be altered by master terminal command, to permit alternate routing in case an execution is unavailable.

Logical network operation

Figure 3 illustrates the message flow from and to a single logical terminal. Inputs from logical terminals are of two general types: *commands* and *messages*. Commands are character strings that instruct the system to carry out some operation on behalf of the entering terminal. Commands have the following format:

/command-verb parameters

For example, the command /IAM LTERM CHARLIE causes the system to associate the logical terminal CHARLIE with the physical terminal from which the command is entered. The system carries out the action called for and responds to the terminal in an appropriate way. In the case of the example command, the response is as follows:

DFS058 IAM COMMAND COMPLETED

For commands that request a display of system status, the response is more extensive.

Messages from logical terminals are character strings that are to be routed to one of two types of destination: application programs and other logical terminals. Destinations may be *local*, i.e., associated with the execution that is controlling the inputting terminal. Or, with the MSC feature, destinations may be *remote*, i.e., associated with some other execution.

A message that is destined for an application program is a *transaction*, and contains a *transaction type code* that determines the program to which it will be routed. Input transactions for local application programs are placed in *transaction queues* where they can be accessed by application programs. Transactions for remote programs are placed in a *link queue* for the logical/physical link over which the transactions are to be routed. The transactions are transmitted when the link is available. A message that is destined for the Interactive Query Facility—which operates in the manner of an application program—is called a *query*. Queries are queued in the same manner as messages for application programs.

A message that is destined for another logical terminal is a *switched message*, and contains the name of the logical terminal to which it is to be routed. A switched message for a local terminal is placed in a *terminal queue* that is associated with the terminal, and is transmitted when the terminal is available. Switched messages for remote terminals are processed in the same manner as transactions for remote programs.

If many successive messages are to be routed to the same destination, the operator may preset the destination with the /SET command. Succeeding messages then need not contain the destination name or code. The normal mode of message entry is restored with the /RESET command.

Messages received from a terminal are immediately acknowledged to the terminal by IMS/VS. The form of this acknowledgment depends on the terminal type. For example, for the 3277 display station, acknowledgment consists of unlocking the keyboard for the operator to enter another message. The acknowledgment is independent of any application program response that may be generated to the message.

Outputs to logical terminals come from several sources: application programs, other logical terminals (i.e., the source of switched messages), and the IMS/VS system itself (e.g., responses to commands). In every case, the output for a logical terminal is first placed in the terminal queue that is associated with the logical terminal, and is transmitted when the terminal is available.

A logical terminal may be set into different operating modes by a command from the terminal. For example, in *exclusive* mode, output to the terminal is restricted to application program responses to messages that are entered from the terminal. Other outputs, such as switched messages, are held in the terminal queue until the exclusive mode is cancelled. In the *lock* mode, all outputs for a designated logical terminal are suspended until the mode is cancelled. This mode may be useful when the operator is entering a batch of transactions, and does not wish to be interrupted by an application program response until the entire batch has been entered.

In addition to inputs from its own terminals, an on-line execution can receive input (messages only) over its physical links with other executions. The destination of this input is determined in the same manner as a logical terminal input. Output messages to a physical link are taken from the associated logical link queue.

Both commands and messages may be segmented. On input, segmenting is normally controlled by operator action, such as the depression of a carriage return key. End of message is signaled by an ENTER key or equivalent. Only the first segment of an input message or command carries the destination name or command identifier. On output, segmenting is controlled by the application program. Segment size is often restricted by device characteristics, e.g., print line width in the case of printers.

Message queues are implemented as Overflow Sequential Access Method (OSAM) data sets on direct access storage. Messages are placed in main storage buffers as they are received, and are written to direct access storage only when buffer space is needed for more current messages. Message queue buffer space is determined by the user at system definition. All input and output messages are logged by the control program to provide message queue recoverability in case of system outage.

Security facilities

To protect against unauthorized use of the system, IMS/VS provides two levels of security, basic security and security tables. At the basic level, certain IMS/VS commands are automatically restricted to entry through the master terminal. Generally, these are the commands that affect the on-line execution as a whole, such as startup, shutdown, and the enabling and disabling of lines and terminals. Attempts to enter such commands through other than the master terminal are rejected.

Basic security may be supplemented with optional security tables that are built with the aid of the Security Maintenance utility program. The security definitions in these tables are put into force at each startup of the system, unless they are overridden by the master terminal operator.

The security tables may contain entries of the following form:

$$\left\langle \begin{Bmatrix} \text{transaction-type-code} \\ \text{command-name} \end{Bmatrix}, \text{logical-terminal-name} \right\rangle$$

This form limits the entry of particular transactions or commands to particular logical terminals. Similarly, with the MSC feature, entries of the form

⟨transaction-type-code, remote-execution-id⟩

may be used to require that particular transaction types originate from particular remote executions. A different set of definitions may be supplied for each logical terminal that is associated with a given physical terminal, thereby giving the physical terminal different security attributes, depending on which of its logical terminals is enabled.

Terminal user access may be further controlled by security table entries of the following form:

$$\left\langle \begin{Bmatrix} \text{transaction-type-code} \\ \text{command-name} \end{Bmatrix}, \text{password} \right\rangle$$

Such an entry requires that transactions or commands of the specified type contain a particular password before they are accepted by the system, regardless of the logical terminal from which they are entered. Passwords are entered immediately following the transaction type code or command verb as shown here:

```
TRAN02(ANNIE) THIS IS A TRANSACTION

/LOCK(ANNIE) CHARLIE
```

The effectiveness of passwords is enhanced by terminal features that inhibit their display as they are being keyed.

Editing and formatting

IMS/VS provides facilities for the editing and formatting of input and output messages. The position of editing operations in the overall message flow is indicated in Figure 3. Certain basic types of editing are included implicitly. For input, basic editing includes the removal of control characters, backspaces, and other characters of no significance to the message destination. Input editing also includes the removal of the password from the message, and the insertion of a destination code if the transmitting terminal is operating in the preset destination mode. Basic output editing includes the insertion of control characters required for transmitting a message to a terminal.

Basic editing can be augmented by user-supplied edit routines that are incorporated into the control program during system definition. Edit routines are entered just prior to basic editing. Separate input and output edit routines can be specified for each BTAM line group or VTAM set of terminals of the same type. Their use on an individual terminal basis may also be specified. User edit routines operate on one message segment at a time, and may be used for such purposes as translating commands and responses from one language to another, substituting one password or destination for another in input messages, and rejecting incorrect input messages.

For certain terminal types (2740 and 2741 data communication terminals, 3270 information display system, 3600 finance communication system), IMS/VS provides a Message Format Service (MFS) for interpreting and generating complex input and output data streams, such as those typically found in display terminal use. Rather than requiring application programs to deal directly with such data streams, MFS converts input data streams into "conventional" message formats of the type described earlier in this section, before handing the message to the application program. Similarly, MFS accepts conventional message formats from application programs and converts them to output data streams required by the receiving terminal. Both conversions are carried out using user-supplied definitions of input and output messages, and definitions of input and output data streams in the form of the displays to be presented to the terminal operator. These definitions are entered into the IMS/VS format library with the aid of the MFS language utility program.

For input conversion, MFS uses a Device Input Format (DIF) that describes the input display as prepared by the terminal operator, and a Message Input Descriptor (MID) that describes the message to be derived from the input display for presentation to the application program. The DIF describes each field in the input display, assigning it a name, a row and column starting position on the display, a length, and other similar attributes. For example,

```
ABLE DFLD POS=(10,14),LTH=6
BAKER DFLD POS=(11,20),LTH=8
```

describes the display fields ABLE and BAKER. The MID describes in sequence each field in the message, giving such attributes as its length, the display field from which it is to be derived, its justification, and fill characters, as follows:

```
MFLD ABLE, LTH=6,JUST=R,FILL=C'0'
MFLD BAKER, LTH=10,JUST=L,FILL=X'40'
```

The MID may also define literal fields to be included in the input message, as follows:

```
MFLD CHARLIE, LTH=5, 'CRAZY'
```

A literal field definition may be used to supply the transaction code in a message so that the operator need not enter it in each message.

Input conversion proceeds by scanning the MID, and—for each field therein—by developing a value for the field either from the corresponding display field or from the specified literal value. Provision is made for defining and generating multiple-segment messages, and for incorporating passwords that are derived from display fields or literals.

For output conversion, MFS uses a Message Output Descriptor (MOD) that describes the message as produced by the application program, and a Device Output Format (DOF) that describes the message as it is to be displayed at the terminal. The content of these definitions is roughly analogous to the corresponding input definitions. Conversion proceeds by scanning the MOD, and—for each field defined therein—moving the field value from the message to the corresponding display field. As in input conversion, literals may be generated in the display. Messages to be converted may be segmented. Output displays may be segmented into *pages*, where a page represents the extent of the message that is visible to the operator at any time. Operator control fields may be defined in input displays, and may be selected by the operator (for example, by light pen) for such purposes as paging through a message, selecting a particular page of a message, and advancing to the next message.

Terminals that are supported by MFS may use either the MFS mode of editing, or the basic mode optionally supplemented by user edit routines. When in the MFS edit mode, the user input and output routines used with basic editing are not called. Instead, MFS provides for separate user-written edit routines to be associated with individual fields and segments of a message. These routines are entered after MFS has completed its editing of fields and segments, and may be used to validate fields, modify fields, and cancel segments or messages with operator notification.

In addition to input and output editing, which is done on a terminal or terminal type basis, IMS/VS provides for an additional edit routine to be used on all switched messages. This routine is entered after the input has been determined to be a switched message.

Network control

The control of communication resources allocated to an on-line execution is achieved through master terminal commands. After each startup, /START commands are used to enable lines, physical terminals, and logical terminals. At any time during operation, these resources may be disabled with /STOP, /PSTOP, or /PURGE commands. The latter commands differ in the manner in which subsequent communication with the affected terminals is treated: /STOP stops all communication; /PSTOP allows output message queuing to continue; and /PURGE allows both output message queuing and transmission to continue. Terminals that have been disabled may be re-enabled by the /START or /RSTART command. The former resets the terminal to a standard mode, and the latter enables it in the mode it attained when disabled.

Summary and concluding remarks

The communication facilities of IMS/VS permit the user to construct networks of on-line system executions, with each execution controlling a configuration of lines and terminals of various types. Facilities are provided for the transmission of commands and responses between a terminal and its controlling execution, and for the transmission of messages between a terminal and an application program or another terminal located anywhere in the network. Communication facilities include security facilities to control terminal access to the network, and editing facilities to simplify the application programming of communication functions. Network resources are controlled from the master terminals that are associated with on-line executions.

GENERAL REFERENCES

1. IBM Corporation, *IMS/VS Version 1 System/Application Design Guide*, Document SH20-9025, IBM Corporation, Data Processing Division, White Plains, New York 10604.

2. IBM Corporation, *IMS/VS Version 1 Installation Guide*, Document SH20-9081, IBM Corporation, Data Processing Division, White Plains, New York 10604.

3. IBM Corporation, *IMS/VS Version 1 Operator's Reference Manual*, Document SH20-9028, IBM Corporation, Data Processing Division, White Plains, New York 10604.

4. IBM Corporation, *IMS/VS Version 1 Message Format Service User's Guide*, Document SH20-9053, IBM Corporation, Data Processing Division, White Plains, New York 10604.

Reprint Order No. G321-5049.