

SH20-9071-2

Program Product

**VS Personal Computing (VSPC)
General User's Guide and
Command Language**

IBM

SH20-9071-2
File No. S370-39

Program Product

**VS Personal Computing (VSPC)
General User's Guide and
Command Language**

Program Numbers 5740-XR5 (OS/VS1 VSPC)
5740-XR6 (OS/VS2 MVS VSPC)

IBM

Third Edition (September 1978)

This edition, as amended by technical newsletter SN20-9317, applies to Release 2 of VSPC, Program Products 5740-XR5 (OS/VS1 VSPC) and 5740-XR6 (OS/VS2 MVS VSPC), and to any subsequent releases unless otherwise indicated in new editions or technical newsletters.

The changes for this edition are summarized under "Summary of Amendments" following the list of figures. Specific changes are indicated by a vertical bar to the left of the change. These bars will be deleted at any subsequent republication of the page affected. Editorial changes that have no technical significance are not noted.

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.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address 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 reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California, U.S.A. 95150. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

ABOUT THIS BOOK

This manual describes the VSPC general commands and the VSPC capabilities they control; these commands and capabilities are those available to the general user. With one exception, such users need not have prior data processing experience; the exception is those users who expect to use the VSPC batch processing capabilities.

VS Personal Computing (VSPC) provides interactive and batch processing capabilities; the three program products run under OS/VS1 (Program Number 5740-XR5), OS/VS2 MVS (Program Number 5740-XR6), or DOS/VS (Program Number 5746-XR3). This manual follows the organization outlined below:

- “VSPC Facilities” introduces VSPC concepts and describes the facilities available to the general problem-solving user. This information is needed for an understanding of the later chapters.
- “VSPC Usage” describes how to log on and off, how to conduct a VSPC terminal session, how to enter each VSPC command, and what results to expect. This chapter is meant to teach general users unfamiliar with computing systems the procedures needed to use VSPC effectively. Many examples of general commands are given.
- “VSPC General User’s Dictionary” gives a formal definition of each VSPC general command and of related VSPC terms. It is a reference guide for VSPC general users who need information on specific points of VSPC operation.
- An Appendix of “Terminal Features,” an Appendix of “How to Produce a Listing of Online-Message Explanations,” and a Glossary are also included.

Several IBM programs can be used with VSPC:

- VS APL (Program Number 5748-AP1)
- VS BASIC (Program Number 5748-XX1)
- VSPC FORTRAN (Program Number 5748-FO2)
- VSPC PL/I (Program Number 5785-EAB)
- VS TSIO (Program Number 5740-XR9)

Information for the general user regarding these programs is contained in the following publications:

- *VS APL for VSPC: Terminal User’s Guide*, Order No. SH20-9066
- *VS BASIC for VSPC: Terminal User’s Guide*, Order No. SH20-9060
- *VSPC FORTRAN Terminal User’s Guide*, Order No. SH20-9062
- *VSPC PL/I Terminal User’s Guide*, Order No. SH19-6048
- *VS TSIO Guide and Reference*, Order No. SH20-9107

This publication describes only those VSPC facilities available to the general user. More advanced information is included in the following manuals:

- *VS Personal Computing (VSPC) Installation Reference Material*, Order No. SH20-9072, provides information on the VSPC supervisory commands, on installing VSPC, and on day-to-day VSPC operation.
- *VS Personal Computing (VSPC): Writing Processors*, Order No. SH20-9074, provides information on designing and writing foreground and auxiliary processors to run under VSPC.

The following publications are referred to in this book:

- *System/370 Reference Summary*, Order No. GX20-1850
- *VS Personal Computing (VSPC) Terminals*, Order No. SH20-9073

Note: VSPC message explanations are contained in a VSPC load module, which can be printed as described in Appendix B.

CONTENTS

About This Book	3
Figures	11
Summary of Amendments	13
VSPC Facilities	15
A VSPC Overview.....	15
Your User Profile.....	17
Your User Number.....	17
Your Workspace Attribute	18
Your Library	18
Your Logon Password.....	18
Your Job Entry Code.....	18
VSPC Commands	19
Command Format	19
VSPC Data	21
Line Numbers.....	21
Using Data.....	22
VSPC Libraries	23
Filenames	23
Private Libraries	23
Project Libraries.....	24
Public Libraries	24
File Types.....	24
VSPC Access Control Features	25
Passwords.....	25
File Protection.....	26
File Sharing	26
VSPC Usage	27
Valid VSPC Terminals	27
IBM 3270 Information Display System.....	27
IBM 3101 Display Terminal.....	27
IBM 3767 Communication Terminal	27
IBM 3770 Data Communication System.....	29
IBM 2741 Communication Terminal	29
IBM 1050 Data Communication System.....	30
CPT-TWX (Model 33/35)	30
Using Your Terminal	30
Current Line Pointer	31
Starting a Session	33
Conducting a 3767 (S/S) or 2741 Terminal Session	33
Correcting and Deleting the Current Line	35
Using the Attention Key	36
Invalid Output Characters.....	36
Conducting a 3270 Display Terminal Session	36
Character Correction	37
Character Insertion	37
Character Deletion.....	37
Line Correction.....	37
Special Requests.....	37
Display Screen Hold.....	38
Invalid Output Characters.....	38
Using Program Function (PF) Keys—PFKEY Command	38

	Displaying PFKEY Settings	40
	Requesting Printer Services	40
	Command Mode Entry and Editing.....	41
	Line Numbered Data Entry.....	42
	Automatic Line Numbering—INPUT Command	43
	View Mode Entry and Editing.....	46
	Entering View Mode—VIEW Command	47
	Changing the Screen Display—VIEW Command.....	47
	Using the Control Area	47
	Message Area	48
	Using the Workspace Display Area.....	48
	Using the Bulk Input Area	49
	Using Local Commands	50
	Order of Processing.....	51
	Cursor Positioning.....	52
	View Mode Processing Procedures	53
	Session Oriented Commands	53
	Changing the Logon Password—PASSWORD Command	53
	Presetting Command Actions—TEXT Command.....	53
	Setting Session Line Length—LINESIZE Command.....	54
	Setting Session Tabs—TABSET Command	55
	Setting Logical Terminal Characters	55
	Character Translation—TRANSLATE Command.....	57
	Checking VSPC Usage—QUERY Command	59
	Changing Your Workspace Attribute—ENTER Command.....	60
	Saving Your Work Automatically.....	60
	Ending a Terminal Session—OFF Command.....	61
	Using VSPC Message Facilities	62
	Sending Messages—SEND Command.....	62
	Receiving Messages—MESSAGE Command.....	62
	Identifying Messages	63
	Handling Workspace Data	64
	Erasing Data—DELETE Command.....	64
	Rearranging Lines—MOVE Command.....	65
	Repetition of Data Lines—COPY Command.....	66
	Reformatting Lines—JOIN and SPLIT Commands	67
	Replacing Data—CHANGE Command	67
	Locating Data—FIND Command.....	69
	Changing Line Numbering—RENUMBER Command	70
	Naming Your Workspace—NAME Command.....	71
	Saving Workspace Contents—SAVE Command.....	72
	Displaying Your Workspace—LIST Command	72
	Erasing an Entire Workspace—CLEAR Command.....	73
	Erasing a Partial Workspace—EXTRACT Command	73
	Using VSPC Libraries.....	74
	Obtaining a File—LOAD Command.....	74
	Combining Files—MERGE Command	74
	Creating and Modifying Files—FILE Command	75
	Assigning Sequence Numbers—NUMBER Command.....	76
	Restricting File Access—PROTECT Command	78
	Extending File Access—SHARE Command	79
	Transferring Files—ACQUIRE and RELEASE Commands	80
	Displaying Library Directories—QUERY LIBRARY Command	81
	Displaying File Information—QUERY FILE Command	81
	Removing Files—PURGE Command.....	82
	Using Foreground Processors	82

Interpreters.....	82
Compilers	83
Setting Processor-name Attribute—ENTER Command.....	83
Writing the Source Program	84
Placing Source in a Library—SAVE Command	84
Compiling and Saving Your Program—STORE Command.....	84
Executing Your Program—RUN Command.....	85
Assigning Unit Numbers—ALLOCATE Command	86
Releasing Unit Numbers—FREE Command.....	86
Processing Command Lists.....	87
Command List Creation.....	87
Command List Execution—RUN Command.....	89
Command List Nesting—RUN Command	90
Processing Batch Jobs.....	90
Programming Considerations for Job Entry.....	91
Submitting Batch Jobs—SUBMIT Command.....	92
Querying Job Execution—STATUS Command.....	93
Obtaining Output at Terminal—LOAD Command	94
Alternate Output Destinations—ROUTE Command	95
Canceling a Batch Job—CANCEL Command.....	96
Erasing Job Output—SCRATCH Command	97
Getting Help in Forming Commands—AID Commands	98
VSPC General User's Dictionary	101
Command Syntax.....	103
Abbreviations of Keywords.....	104
Command Continuation.....	104
Comments	105
Format Notation	105
COMMAND-NAME	105
Operands	105
Explanatory Information	106
ACQUIRE Command	107
AID Commands.....	107
ALLOCATE Command.....	110
BACKSPACE Command	111
CANCEL Job Entry Command	112
CHANGE Command	113
CLEAR Command	115
COPY Command.....	115
DELETE Command	116
ENTER Command	117
EXTRACT Command.....	118
FEATURE Command	118.1
FILE Command.....	119
File Organization	121
Sequential Files	121
Direct Files	121
Object Program Files.....	121
Undefined Files	122
External VSAM Files	122
Filenames	122
Special File CONTINUE	123
FIND Command.....	124
FREE Command	125
HARDCOPY Command	126
INPUT Command.....	126

Jobnames	128
JOIN Command.....	129
KEY Command	129
Libraries.....	130
Private Libraries.....	130
Project Libraries.....	130
Public Libraries	131
Line Entry—Command Mode.....	132
Line Entry—View Mode.....	132
LINESIZE Command.....	134
LIST Command	135
LOAD Command—to Retrieve VSPC Files.....	136
LOAD Job Entry Command—to Retrieve Output	136
Local Commands—VIEW Mode Only	138
LOCATE Command	140
MERGE Command	141
MESSAGE Command.....	143
MOVE Command.....	144
NAME Command.....	145
NEWLINE Command.....	146
NUMBER Command	147
OFF Command.....	149
PASSWORD Command	149
Passwords	150
Password Assignment.....	150
Password Prompting.....	151
PFKEY Command.....	152
PROTECT Command	153
PUNCH Command.....	155
PURGE Command.....	156
QUERY Command—to Display File Status	156
QUERY Command—to Display User's Status.....	160
RELEASE Command.....	163
RENUMBER Command	164
ROUTE Job Entry Command.....	165
RUN Command.....	166
SAVE Command	167
SCRATCH Job Entry Command.....	168
SEND Command	169
SHARE Command	170
SPLIT Command.....	171
STATUS Job Control Command.....	172
STORE Command.....	173
SUBMIT Job Entry Command.....	174
TABSET Command.....	176
TAPE Command	177
TEXT Command	178
TRANSLATE Command	178
VIEW Command.....	179
VSPC (Logon) Command.....	180

Appendix A. Terminal Features.....	183
Appendix B. How to Produce Listing of Online-Message Explanations	185
Glossary.....	187
Index	191

FIGURES

Figure 1.	VSPC Overview	17
Figure 2.	3270 Keyboard Layout	28
Figure 3.	3767 General Keyboard Layout	28
Figure 4.	3774 and 3775 Keyboard and Operator Panel	29
Figure 5.	Format of VSPC Examples	30
Figure 6.	Information Assumed in Examples	31
Figure 7.	Switched Line Terminal Connection Procedures	33
Figure 8.	Command Mode 3270 Screen Display Format	41
Figure 9.	View Mode 3270 Screen Display	46
Figure 10.	Using MERGE with OVERLAY	75
Figure 11.	Creating the PERSLIST Command List	88
Figure 12.	Executing the PERSLIST Command List	89
Figure 13.	Developing COBOL Batch Programs	92
Figure 14.	VSPC Command Functions	102
Figure 15.	Command Format Presentation	105
Figure 16.	IBM Program Products and Valid Processor-names	117
Figure 17.	Library Types and Access Control Commands.....	131

SUMMARY OF AMENDMENTS

February, 1980

Service Changes

- Support for the 3101 Display Terminal
- Clarification of View Mode entry and editing

September, 1978

VSPC Release 2 for OS/VS

New Programming Features

- Support for 3270 Display System devices 3276/3278
- Improved terminal support
 - View Mode entry and editing
 - Program Function (PF) keys
- Current Line Pointer support
- Error Message clarification
- New Commands: PFKEY, VIEW, NEWLINE, BACKSPACE, COPY, LOCATE, EXTRACT, JOIN, SPLIT, TEXT, RELEASE, ACQUIRE
- Improved Commands: LINESIZE, TABSET, TRANSLATE, HARDCOPY, INPUT, LIST, CHANGE, FILE, PUNCH

September, 1977

VSPC FORTRAN Link Processor (OS/VS Only)

New Programming Feature

The VSPC FORTRAN subroutine link processor has been added as a foreground processor and the *'string'* operand added to the STORE and RUN commands.

Service Change

The logon procedure with 3270 terminals has been clarified.

VSPC FACILITIES

VSPC (VS Personal Computing) is designed to help you solve problems quickly and easily using a computer terminal. From the terminal, you “talk” to VSPC and tell it what you want the computer to do. VSPC then translates your requests into computer language and gets the computer system to do the work. When the work is completed, VSPC tells you whether the results were successful or not, and if not why not.

Before you begin using VSPC, there are fundamental VSPC concepts you must understand. In this chapter, these concepts are explained. First, there is an overall description of VSPC and how it relates to you, the terminal user. Then there is a more detailed description of each aspect of VSPC you will be using in your day-to-day work: your user profile, VSPC commands, VSPC data, VSPC libraries, and VSPC access control features. This necessary preliminary information is informative only—the next chapter will tell you how to use each of the features described in this.

A VSPC Overview

When you talk to VSPC, you tell it to perform a specific processing request, which VSPC then performs—or executes—for you. When VSPC has completed executing the request, it tells you whether or not it was able to complete the request correctly. Then you repeat the process by telling VSPC to perform another specific processing request, and VSPC again responds. Thus, VSPC processing is *conversational* or *interactive*. In VSPC, interactive processing, also sometimes called *online* processing, takes place in an area known as the *foreground*. Foreground processors—not part of VSPC itself, but designed to work with VSPC—make use of the VSPC interactive facilities to do work for you.

Interactive processing contrasts with conventional *batch* processing—which may also be available to you through VSPC. In batch processing, you tell the computer system to perform a whole series of actions in one complete set of instructions. The computer then tries to execute the whole series, and if it can't perform some or all of your requests, it records one or more error messages. The results of the entire series of requests—and the error messages—are then returned to you as a single unit. Each batch job is processed as time permits; thus, the results may not be returned to you for some time. As you can see, such processing cannot be conversational. Batch processing, also sometimes called *offline* processing, in VSPC takes place in an area known as the *background*. The VSPC job entry facilities let you use the VSPC batch processing facilities.

VSPC allows you and a number of other terminal users to do conversational processing at the same time. That is, you can all talk to VSPC and get VSPC's answers simultaneously.

Because VSPC must have a way of identifying your work, you are assigned a *user number*. When you log on to VSPC (that is, when you begin a terminal session), you must always identify yourself with this user number.

Your user number is assigned by the *VSPC administrator*, a user who has the responsibility for the VSPC facilities available to you. The VSPC

administrator assigns your user number and specifies other characteristics of the work you'll be doing.

When you log on, VSPC furnishes you with a *workspace*. This *workspace* is an area you can use much as you use a blackboard or scratchpad in ordinary problem solving. You can write down solutions, erase them, modify them, add to them, subtract from them, write in reminders to yourself. If the problem you are working on needs only a one-time solution, you can execute it, and then erase it from your workspace. Alternatively, if the problem solution will be needed again later, you can copy the solution and save it for future reference.

You save the copy in a user *library*; when your user number is assigned, VSPC automatically assigns you at least one such library, and may make more than one available to you. VSPC uses one general library (something like a large metropolitan library system) that contains a library for each user. Each VSPC user library is something like an ordinary library—you save *files* (equivalent to the books in an ordinary library), each identified by its *filename* (equivalent to the book title), in a specific library (something like a branch library in a library system). VSPC keeps a directory (equivalent to a card catalog) to each *file* in a library.

In both the workspace and the libraries, you can place *data*.

Data can be anything you want it to be—a specific problem solution you have written down, information you will use to test your problem solution, lists of computer instructions or statements, even a set of random numbers or letters.

When you enter *data* from your terminal—you type it in from the terminal keyboard—you are placing the *data* in your workspace. Once the *data* is there, you can edit it—add lines, delete lines, delete characters, move lines from one area to another, change words and phrases, and so forth. Any time you want, you can print or display it at the terminal, save it in a library, and/or erase it.

What's more, if your data should not be used by anyone else, you can specify a *file password*, known only by you, which must be supplied before the data can be retrieved from the library. You can assign a *file password* to each of your files. In addition, your VSPC administrator may assign you a *logon password* which you must supply when you log on to do work; this is to prevent unauthorized access to your library. You can change either kind of password whenever necessary.

You do all your work through VSPC *commands*. *Commands* tell VSPC to perform specific actions. There are commands that help you use the terminal itself (for example, tell the computer what terminal tab settings you have set), others that work on the workspace (for example, erase any data in it); there are commands to edit your data; library commands; commands that let you do batch processing; and other types of commands as well.

In the following sections, you'll learn more about these subjects—your user profile and its relation to your workspace, VSPC commands, data, libraries, and VSPC access control. Figure 1 illustrates the logical relationships of VSPC components.

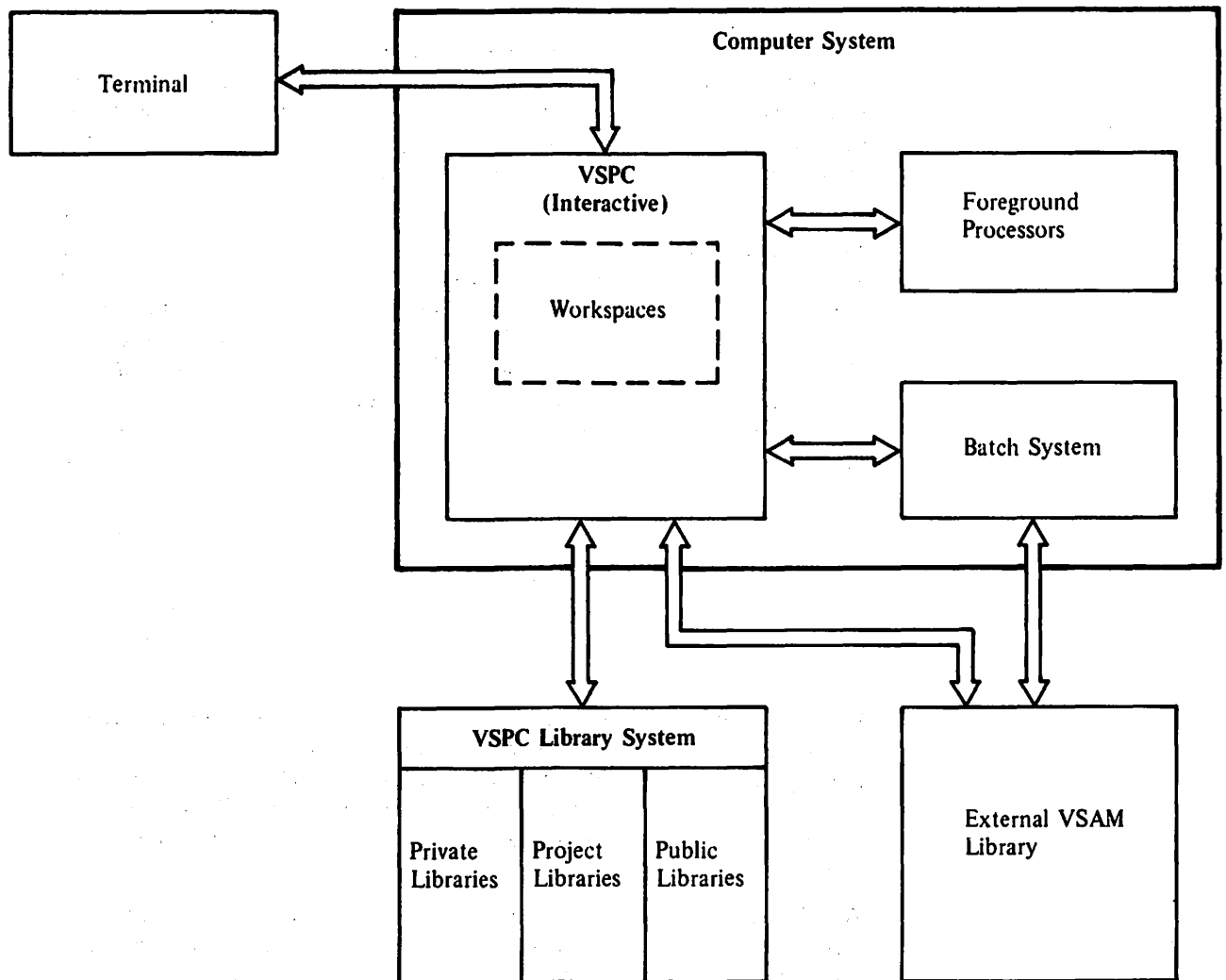


Figure 1. VSPC Overview

Your User Profile

Before VSPC can recognize you and allow you to do work, your VSPC administrator must introduce you to VSPC through a *user profile*. Your *user profile* defines you to VSPC through:

Your User Number

Your user number identifies you to VSPC. Whenever you log on, you identify yourself by typing your *user number*, which is one through seven digits long, and is a unique value.

Your Workspace Attribute

Your workspace *attribute* tells VSPC what kind of work you'll most often be using your workspace for. Your workspace *attribute* can be:

DATA: which specifies you'll be entering simple information. The information can be whatever you choose it to be.

CLIST: which specifies you'll be entering VSPC commands in the form of data—you'll learn about it in the "Processing Command Lists" section of the next chapter.

processor-name: which names a foreground processor available at your installation and specifies that you'll be writing programs to be treated as data input to the processor. A foreground processor allows you to write programs in easy-to-understand language (called *source programs*) that the processor then translates into computer language programs (called *object programs*) and executes. There's a discussion of foreground processors and how to use them in the "Using Foreground Processors" section of the next chapter.

Your Library

Information about the libraries you'll use is included—what kind of library is provided for your use, how much computer storage you can use for it, and what additional libraries are available to you.

Your Logon Password

A logon password can optionally be assigned you. If one is assigned, you then must supply the correct password each time you log on to do work. If you wish, you can change your logon password during your terminal session; you must then supply the new password the next time you log on.

Your Job Entry Code

When you are assigned a job entry code, you are allowed to submit batch processing jobs through VSPC facilities. Your job entry code is unique, and identifies you to the batch processing system. (Details of job entry are discussed in the "Processing Batch Jobs" section of the next chapter.)

Once your VSPC administrator has introduced you to VSPC by defining your user profile, you can log on to do work. You do work by specifying VSPC commands.

VSPC Commands

A VSPC *command* is a word that tells VSPC to perform a specific action. The kinds of actions the commands perform can be thought of in groups of processing capabilities as follows:

Terminal-Oriented Commands help you use the terminal for communicating with VSPC. Using these commands you can log on and log off (start and end a terminal session), change your logon password, change the line length and tab settings, and define logical control characters for newline, backspace, and tab.

Message Commands let you send messages to the operator or users on other terminals, and allow you to prepare your terminal to accept or refuse messages from others.

Data Editing Commands let you add lines to the workspace, change lines in the workspace, delete characters or lines, replace characters or lines, move or copy lines within the workspace, join or split lines, change line numbering, merge a file with workspace, locate occurrences of a group of characters, and refer to lines by means of a current line pointer.

Display-Oriented Commands let you use an IBM 3270 Information Display System to directly modify the contents of your workspace as displayed on the screen, and to define program function keys for a wide variety of functions and to get a hard copy print of the screen image.

Workspace Oriented Commands let you name your workspace contents and save them in a file, assign attributes to a workspace, load the workspace from a file, and delete workspace contents when they are no longer needed.

Library Management Commands let you create and purge files in a library, specify that other users can or cannot read a file, specify that the file cannot be changed, and allow the transfer of ownership of a file. You can also display the library directory.

Job Entry Commands let you submit jobs for batch processing, to query their status, to cancel them, to specify that job output is to be sent to some specific destination, and to erase job output once it has been produced. (Job entry commands are not always available for your use; check with your VSPC administrator.)

AID Commands let you receive online explanations of VSPC messages, and online prompting in the formation of valid VSPC commands. (AID commands are not always available for your use; check with your VSPC administrator.)

Command Format

Each *command* has a format similar to every other. The first word in a command is always the *command name*, a familiar English word that specifies one particular action to be performed. The command name is optionally followed by one or more *operands* that provide additional information or that modify the action of the command. For example, if you type:

```
query library
```

you are telling VSPC to display all filenames in your own library at your terminal. Then, if you type:

```
query library (100)
```

you are telling VSPC to display all filenames in library 100 at your terminal. Then, if you type:

```
query library (100) own
```

you are telling VSPC to display only the filenames of your own files in library 100.

Note that the QUERY LIBRARY command always means that *some* filenames are to be displayed at your terminal; an operand (or its absence) defines *which* filenames are to be displayed.

Each *command* begins on a line by itself, and the first meaningful characters on the line must be the command name. If anything but spaces, commas, or tabs precedes the command name on the line, the command cannot be recognized by VSPC.

Command Operands

Note, too, that operands can be of two kinds, variable or keyword:

variable operands: which supply VSPC with information that you, the terminal user, determine. In the preceding examples, 100 is this kind of operand; the library number is one that you yourself choose.

keyword operands: which define a predetermined action to VSPC. In the preceding examples, OWN is a keyword operand—it has been predefined to mean that VSPC is to display only filenames of your own files at your terminal.

Some commands use only variable operands; other commands use both variable and keyword operands. Only a few commands use only keyword operands. Some commands use no operands at all.

Punctuation in operands has special meanings, as described in the following paragraphs:

- Parentheses (()) optionally enclose variable information that governs the action of the keyword operand (as illustrated in the “Displaying File Information—QUERY FILE Command” section of the next chapter).
- A colon (:) placed between two numbers specifies a range of numbers (as illustrated in the “Printing Your Workspace—LIST Command” section of the next chapter).
- Single quotation marks (') precede and follow groups of characters—called character strings—which are to be treated by VSPC exactly as specified (as illustrated in the “Sending Messages—SEND Command” and “Receiving Messages” sections of the next chapter).

Command Abbreviations

In a command, both the command name and the keyword operands can be abbreviated to their leftmost significant characters. Among command names, the characters must be unique among all command names (that is, the SAVE, SHARE, and STORE commands can be abbreviated respectively as SA, SH, and ST). A keyword operand must be uniquely spelled within a command (that is, the OWN operand of the QUERY command can be abbreviated as O or OW).

Successively longer abbreviations, adding characters at the right are permitted. For example, the PROFILE operand of the QUERY command can be abbreviated as P, PR, PRO, PROF, PROFI, and PROFIL.

An easy rule to remember is that in VSPC you can always be sure that any command name or keyword operand can always be abbreviated to its first three characters.

Separators

In the preceding examples, you'll notice that the operand is always separated from the command name by a space. This is because VSPC requires a *separator* between each operand and command name: valid separators are blanks, commas, and tabs. What's more, VSPC recognizes one or more separators as logically one separator only. No matter how many you type, VSPC recognizes any combination of separator characters as one separator. For example, each of the following commands has exactly the same meaning to VSPC:

```
query profile
query profile
query,profile
query(t) profile
query, (t)profile
```

Note: (*t*) in the preceding examples represents one stroke of the terminal keyboard tab key. The tab is a valid separator only if the TABSET command has been specified to either set session tabs or to set logical tabs with the NOEXPAND option.

VSPC Data

When you're typing at the terminal, you can enter commands, as previously described, or you can enter *data*. Data can be anything you want it to be: a specific problem solution, testing information, a computer program, anything at all.

In VSPC, the most frequently-used forms of data are:

- A computer program you write for later execution (there are many types of such programs, some of which you'll read about in the next chapter)
- A command list you'll execute later (command list processing is described in the next chapter)
- Information you'll use during the processing of a computer program

Line Numbers

Data you enter from your terminal always begins with a *line number*—that is how VSPC recognizes data as different from commands. The line number tells VSPC that whatever follows on this line is data, and is to be placed in your workspace. The line number identifies this line of information with one through five digits, and makes it possible to refer back to this particular line at any future time—to display it, change it, replace it, or to delete it. For example, if you type:

```
query library (100)
```

that is a command; VSPC recognizes it as such, and performs the indicated action. However, if you type:

```
080 query library (100)
```

VSPC recognizes 080 as a line number, and treats the QUERY command that follows as data, placing it in the workspace. For example, all of the following lines are treated as data by VSPC:

```
100 print i
080 query library, 100
120 80 100 120 140 160 180 200
250 TYPE: The quick brown fox jumps over the lazy dog
```

In VSPC you create line numbers in one of two ways:

- Typing each line number and then typing the line of information (you can type the line numbers in any numerical order, and VSPC will automatically rearrange the data lines in ascending numerical order for you). If your line numbers contain fewer than five digits and your first data character is numeric, you must type a space between the line number and the data.
- Asking VSPC to generate line numbers for you (see “Automatic Line Numbering”).

Using Data

Once you’ve created your data, you can use it as you please: you can use it immediately and then erase it, or you can save it in a library under a unique *filename*, or you can use it immediately *and* save it for future reference. If you want to, you can erase it without using it at all.

There are three important things you should always keep in mind when you are creating data:

- During one terminal session, unless you have explicitly erased the data from your workspace, it still exists in the workspace. Suppose, for example, you have created and saved one group of test data and now want to begin creating a second set. If you don’t erase the first set of data before you begin work on the second, you will simply add the second set of test data to the first (and, depending on the line numbers you’re assigning, at the beginning, middle, or end of the first set). This may not be what you want to do at all.

At the end of your terminal session, unless you specifically tell VSPC to save your current data, VSPC will erase it when you log off. That is, if you do not save your data in a library during the session and do not explicitly specify that it should be saved when you log off, the data will not be available the next time you log on.

- If you are working at the terminal and for some reason you’re disconnected, VSPC always saves your current data for you—unless the shutdown is due to actual machine failure (in which case VSPC still makes the attempt). If VSPC can save your data, then when the computer system begins working again, you’ll be able to retrieve the data you were working on.

VSPC Libraries

In VSPC, your user profile always defines a library for your own use; your library has a *library number* (which is the same as your user number), and you are the *library manager* of this library. Your own library is always available to you, and other libraries may also be available. (VSPC uses the Virtual Storage Access Method (VSAM) for all libraries.)

Filenames

When you save new work in a library, you become the *owner* of the file you create, and you must always specify a unique *filename* for it. When you specify a filename, you must specify the following components:

1. Optionally, the *library number* (which is always the one-digit through seven-digit user number of the library manager). When you're referring to a file in your own library, you can omit the library number.
2. The *name*, which is required, and which must be a unique identifying name within this library; a *name* is one through eight characters long—the first character being alphabetic, and the following characters being A through Z, 0 through 9, or \$, #, or @. Some foreground processors use a different naming convention; check the appropriate terminal user's guide.
3. Optionally, the *file password*. (It is described in the "VSPC Access Control Features" section.)

Every file in your library has an *attribute* which you yourself control. The file attribute is the attribute your workspace had when you placed the file in the library. Because you yourself can change your workspace attribute, the file attribute is not necessarily the attribute your workspace has when you log on.

Three kinds of libraries—private, project, or public—can be defined in VSPC, and your VSPC administrator may make you the library manager of any kind.

Private Libraries

If your library is a private library, it is meant primarily for your use only. You are both the *library manager* and the *owner* of every file. You can, if you wish, specify that some files can be read by other users; you can also specify that some of your files cannot be changed even by you yourself.

In addition to your own library, your VSPC administrator may have made a project library available to you as well. The public libraries are available to you, as they are to every other terminal user.

Project Libraries

If your library is a project library, then a specific group of VSPC users is allowed to read files from your library. Your VSPC administrator can define your project library in either of the following ways:

1. Other project users are allowed to place files in the library; each such user is the *owner* of the files he places within it. You are the *library manager*; however, you are the *file owner* only of those files you yourself place in the library. Only the *owner* of a file can modify it; however, ownership of project files can be transferred among users. Only the *owner* or you, the *library manager*, can delete it. Such libraries are known as *noncontrolled project libraries*.
2. Other project users can read files in the project library; they cannot place files into it. Thus, you are the *owner* of every file, and so only you can modify or delete anything in the library. Such libraries are known as *controlled project libraries*.

If yours is a project library, your VSPC administrator may also make another project library available for your use.

Public Libraries

If your library is a public library, then every VSPC user can read files from your library. Public libraries can be of the same two kinds as project libraries, and the same rules apply to file modification and deletion except that ownership of files cannot be transferred.

If yours is a public library, your VSPC administrator may also make a project library available for your use.

File Types

In any VSPC library you can place any of the five types of VSPC files: sequential, direct, object program, undefined sequential, or undefined direct. Each type is described in the following paragraphs.

Sequential Files: are ones in which foreground processors always process the records in the order of their line numbers. The records are placed in the file in line number order, and foreground processors always retrieve them from the file in that order. However, you can edit the lines of a sequential file in any order. Records can be of varying lengths, and the increment between line numbers is of no importance.

Direct Files: are files in which foreground processors can process a specific record through a reference to its line number. That is, for example, your program can process record 200 before record 149—and these can be the only two records it processes in the entire file. Your program can also process records sequentially in line number order. You yourself can edit the lines of a direct file in any order. The records are placed in the file in line number order and the line numbers must increment by one (that is, line numbers must be 1, 2, 3, 4, 5, etc.). All records must be the same length.

Object Program Files: are produced by the foreground processors. Such files consist of executable machine instructions. You can use such a file to perform computations and to process data, but you cannot edit it or use it for any other purpose.

Undefined Files: are special purpose data files created by a foreground processor. Only an object program can process or edit such files. Undefined files can be either sequential or direct.

Other Files: In addition, external VSAM files (that is, VSAM files outside of the VSPC library) may be available for your use. Your system administrator can tell you if you can use such files.

VSPC Access Control Features

VSPC access control features help you keep unauthorized people from using your VSPC user number and/or reading your library. They also make it possible to share files from your library with those who have need to use them. VSPC controls access through passwords, file protection, and file sharing.

Passwords

A *password* is a group of characters you can use to limit access to VSPC itself or to your files. There are two kinds of passwords—*logon passwords* and *file passwords*. Both kinds of passwords have the same rules of formation: they can be from one through eight characters in length; they can be made up of any combination of alphabetic characters and numerals 0 through 9. You can specify passwords in lowercase letters; they'll always be translated to CAPITALS. For example, if you specify:

TABOO02

or

taboo02

as a password, VSPC treats it as TABOO02. Note that the letter O and the numeral 0 are recognized as separate characters.

VSPC never prints your *password* legibly at your terminal, but you yourself can choose to enter it legibly.

Logon Passwords

The VSPC administrator can assign you a *logon password* in your user profile. You must provide this password before you are allowed to log on to work with VSPC. Once you have been given a *logon password*, you can change it during a terminal session. The next time you log on, you'll be required to provide the new password, not the original one. You can also give yourself a password if one hasn't been assigned to you.

File Passwords

You yourself can specify *file passwords* when you are saving files for future reference. You add the password at the end of the *filename*. Any time you retrieve the file, you must first supply the correct file password before VSPC lets you access it.

File Protection

You can specify *file protection* for any file you own. That is, you can limit access to the file, either by other users or even by you yourself.

You can specify that no other terminal user can read, display, or copy your file. That is, even if other users can retrieve the file for use as a source program, they still cannot find out what is in it.

You can also specify that you yourself cannot modify a specific file. Other users can still read it and use it for processing as you can.

When both forms of file protection are in effect, others may be able to use the file as a source program, but only you yourself can read it; you cannot change it or delete it.

File Sharing

Files in a VSPC private library can ordinarily only be read by the library manager (who also is the owner of every file in the library). Similarly, files in a VSPC project library can only be read by members of the project.

If you own files in either type of library, VSPC lets you specify that files you choose to share can be accessed by any user. You can also specify that files you previously shared can be returned to their original unshared state.

If you own a file in a noncontrolled project library, and you are not the library manager, you can make the file available so that another user can acquire it. That user can then access and update the file.

Note that files in a public library are always shared by all users.

VSPC USAGE

Now that we've described the VSPC facilities, let's explore how you can use them. This chapter describes briefly, the valid VSPC terminals, how to log on, how to conduct a terminal session, and how to log off. In addition, we'll give examples of ways to use each of the VSPC general commands.

Valid VSPC Terminals

The terminals you can use with VSPC are described in the following paragraphs. For each terminal there are several different keyboards available, and for each some of the features may vary. The variations are described in the *VS Personal Computing (VSPC) Terminals* manual. Your VSPC administrator will tell you which terminals your organization uses.

Note: The valid VSPC switch settings for the terminals described below are shown in Appendix A.

IBM 3270 Information Display System

The IBM 3270 Information Display System can be one of several display terminals—each has a display screen, a typewriter-like keyboard, and other auxiliary equipment. Optionally, a printer and/or a light pen may be attached to the terminal. A cursor (a one-character underline) indicates the screen position you are using—it is easily moved about for character correction, etc. The general keyboard layout for the 3270 is shown in Figure 2.

VSPC uses two special screen formats—Command Mode format (shown in Figure 8), and View Mode format (shown in Figure 9). How to use both is explained later in this chapter.

IBM 3101 Display Terminal

The IBM 3101 Display Terminal is a compact, mobile, desk-top terminal, which includes a 12-inch, 1920 character screen that can be tilted and swiveled; a small logic unit that serves as a base for the screen; and a separate 87-key keyboard.

IBM 3767 Communication Terminal

The IBM 3767 Communication Terminal is a convenient desk-top terminal that has a typewriter-like keyboard. When you are entering input, the print mechanism prints only in a forward direction; when it is printing computer output, however, the mechanism can print alternately forward and backward (which speeds up the process). There is an audible alarm that warns you when you are approaching the end of an input line or when system problems occur. A three-character indicator shows the next print position in the input line, or (when an error occurs) the cause of an error condition. Some controls on the 3767 vary, depending on whether you are in start/stop (S/S) mode or in synchronous data link control (SDLC) mode; your VSPC administrator will tell you which modes you can use. The general layout of the 3767 terminal controls is shown in Figure 3.

Typewriter Keyboard

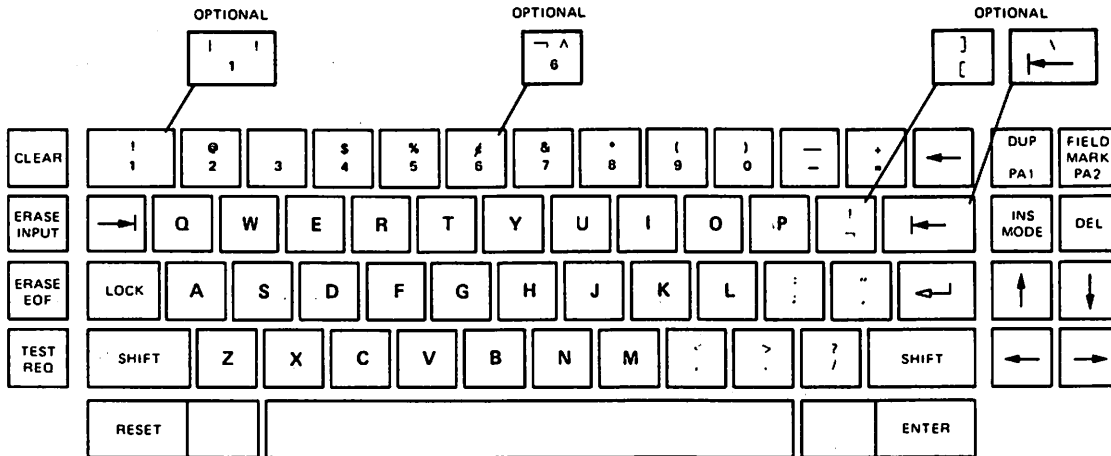
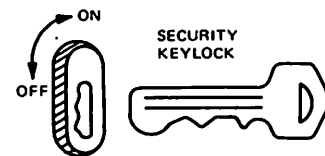
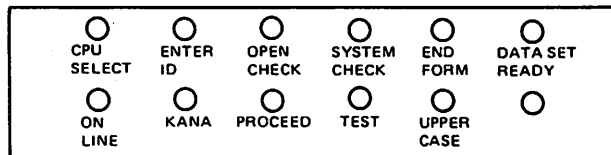
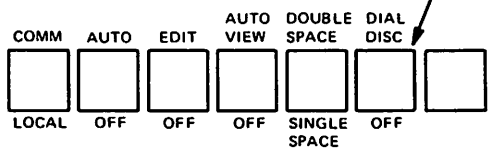


Figure 2. 3270 Keyboard Layout

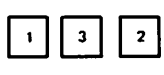
LIGHTS



SWITCHES



COLUMN INDICATOR



SWITCHES

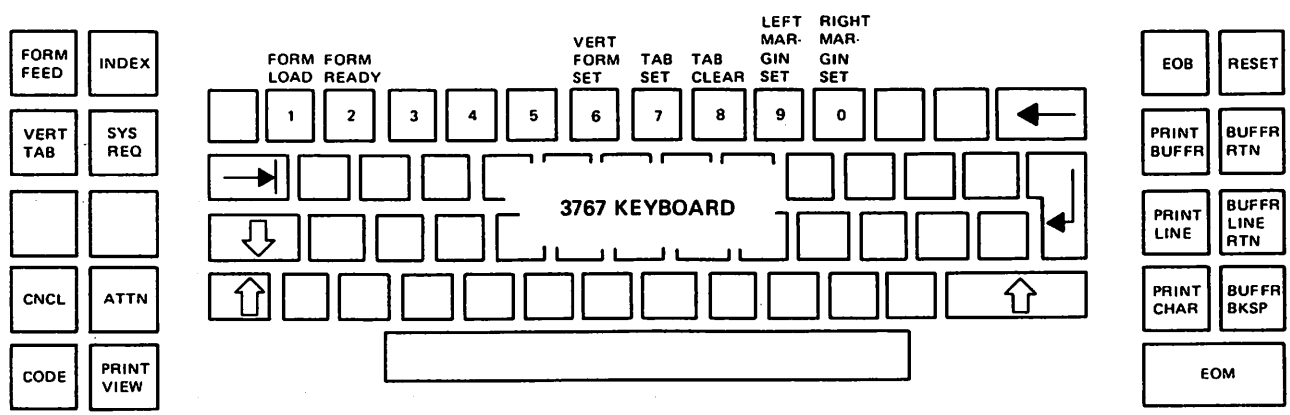
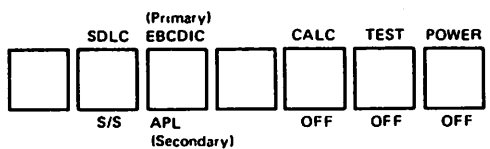


Figure 3. 3767 General Keyboard Layout

IBM 3770 Data Communication System

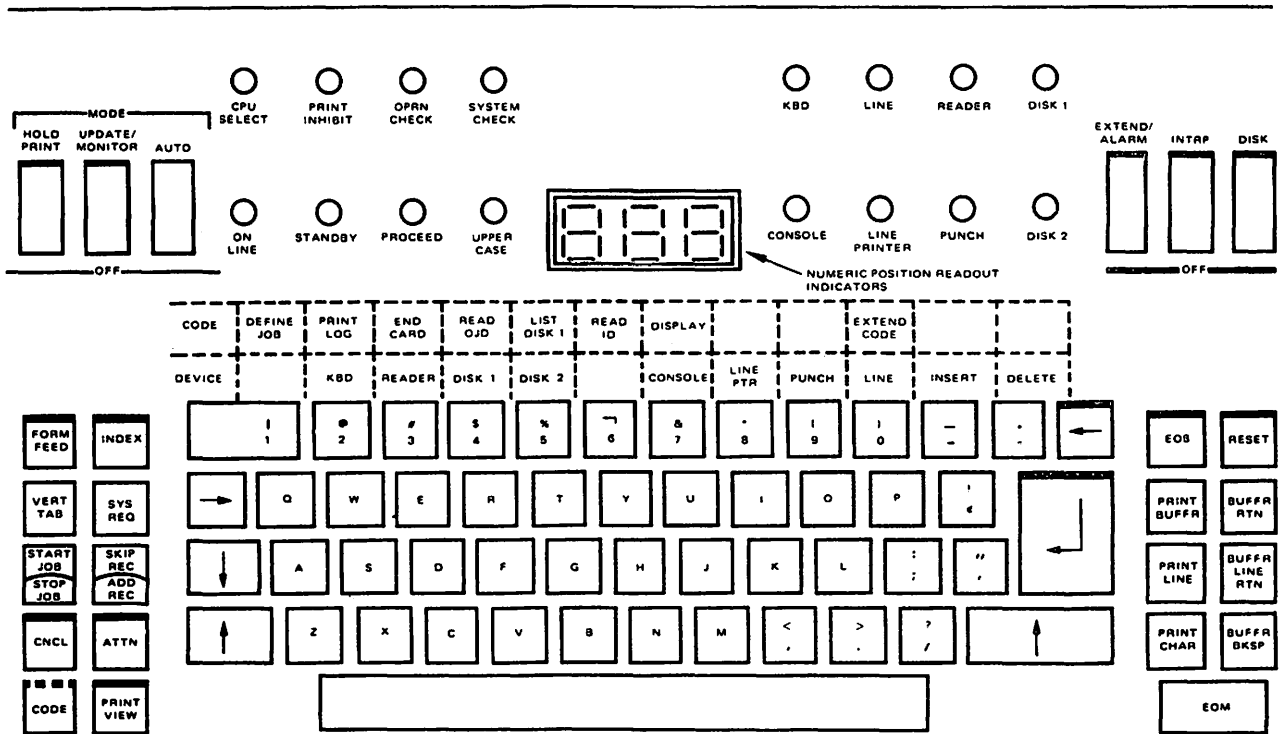
The IBM 3770 Data Communication System is a family of multipurpose keyboard/prINTER terminals and attachable I/O devices. The variety of options, features, I/O devices, and speeds offered by the 3770 terminals and other attachable devices permit configuration of a data communication system for many applications.

The IBM 3771 and the IBM 3773 are multipurpose terminals designed for a range of data-entry, inquiry, and remote printing applications. The 3773 can perform all the same operations that the basic 3771 can perform, and also provides integrated Diskette storage. The IBM 3774 and IBM 3775 each combines all the features and capabilities of the 3771 and 3773 into one versatile multipurpose terminal; and in addition, it offers higher speed I/O, additional Diskette storage capacity, and offline data manipulation capability.

VSPC makes use of only the terminals themselves in online interactive mode using the SDLC method of communication. The keyboard layout for the 3774 and 3775 is shown in Figure 4.

IBM 2741 Communication Terminal

The IBM 2741 Communication Terminal looks like and acts like a conventional IBM SELECTRIC® typewriter with two added controls that allow computer connection.



This figure shows the EBCDIC keyboard arrangement for use in the U.S. and Canada.

Figure 4. 3774 and 3775 Keyboard and Operator Panel

IBM 1050 Data Communication System

The IBM 1050 Data Communication System has a printer-keyboard that resembles a conventional IBM SELECTRIC ® typewriter, plus special keys for computer connection.

CPT-TWX (Model 33/35)

The CPT-TWX Models 33 and 35 include a control panel, a print mechanism, and a keyboard. All alphabetic characters are capital letters; a special key generates control characters. The control panel is used to edit and transmit lines.

Using Your Terminal

The following paragraphs describe how to log on to VSPC, how to enter commands and data, how to edit them, how to set logical computer functions to match the physical ones on your terminal, and how to log off.

VSPC accepts commands entered at your terminal typed in any combination of lowercase and capital letters; before they are acted upon, VSPC translates them into capital letters. Thus, if you enter any of the following:

```
save 200 quotient
SAVE 200 QUOTIENT
save 200 QUOTIENT
SAVE 200 quotient
Save 200 Quotient
```

or any other such combination, VSPC considers them to be equivalent to:

```
SAVE 200 QUOTIENT
```

When VSPC responds to a command, the message is always displayed in capital letters, for example:

```
READY
```

Thus, so that you can distinguish easily between what you enter and what VSPC prints, we'll present all examples as shown in Figure 5.

What you enter is shown in lowercase letters.

Every command ends when you press the ENTER key or the carrier return (*cr*) key.

```
save 200 quotient
ENTER PASSWORD
```

The normal VSPC response is shown in capital letters—as it is displayed at your terminal.

Figure 5. Format of VSPC Examples

Unless otherwise stated, each example is valid for the 3270 display terminal. For each function described, the equivalent for other terminals is summarized in Appendix A.

In the examples used throughout the rest of this chapter, your user profile and other VSPC information are assumed to be those shown in Figure 6.

Your User Profile	
Your user number (usernum)	12345
Your logon password	TABOO
Your library type	private
Your assigned project library	22222
Your content attribute	DATA
Your job entry code	C12345
Other Information	
Public Libraries available	100 200 300
Logon message	YOU MAY BEGIN MAKING ENTRIES

After logon, you've typed in 50 lines, numbered 10 through 500, incremented by 10.

Figure 6. Information Assumed in Examples

Current Line Pointer

One of the lines in your workspace is considered your current line. VSPC maintains a current line pointer to this line. The current line pointer changes with the commands you issue during your editing session. (Commands that use line numbers act on your current line when you enter an asterisk (*) in place of a line number.)

As you enter data into your workspace, the current line pointer points to the last line entered. The pointer can be changed when you issue the CHANGE, CLEAR, COPY, DELETE, EXTRACT, FIND, INPUT, JOIN, LOAD, LOCATE, MERGE, MOVE, or RENUMBER commands.

When you begin an editing session, the current line pointer points to the first line in your workspace. As you issue commands referring to different lines, the current line pointer is updated to the line you're working on.

When you're editing, there will be times when you want to find the current line (it may not be displayed on your screen). You can use the LOCATE command to do this; you type

```
locate
00010
```

(This example assumes that NOTEXT—which tells VSPC to display only the line number—is in effect.)

The response tells you that 00010 is your current line. When you're using a display terminal in View Mode, this changes the screen display so that the current line is the first line (highlighted) at the top of the screen.

Changing The Current Line Pointer – LOCATE Command

You can also use the LOCATE command to change the current line pointer. For example, if your current line pointer is 00050 and you want to change it to 00030, you can issue:

```
locate 30
00030 X + Y = Z
```

(This example and those that follow assume that TEXT— which tells VSPC to display the complete line—is in effect.)

Alternatively, if your current line is 00050, you can use a relative line number to move it back two lines:

```
locate -2
00030 X + Y = Z
```

If you know that a line contains a specific group of characters but you don't know the line number, you can still reposition the current line pointer using the LOCATE command and a contextual search:

```
locate 'x + y = z'
00030 X + Y = Z
```

If you know that the desired characters are contained in a specific part of the workspace, you can limit the contextual search:

```
locate 20:100 'x + y = z'
00030 X + Y = Z
```

Note: If more than one line in the area you're searching contains the characters you're looking for, the current line pointer is set to the first line.

Starting a Session

Before you can log on to VSPC you must make contact with the computer. The procedure you use varies depending on whether the connection is through a leased line or a switched line. Your VSPC administrator will tell you which type of connection you're using.

If your connection is by a leased line, you can make contact with VSPC as soon as you turn the power switch on.

If your connection is by a switched (telephone) line, then you must take the additional steps shown in Figure 7 before you can log on.

Telephone Data Set	Acoustic Coupler
1. Press TALK button.	1. Make sure coupler is: connected to power, turned off, connected to terminal.
2. Remove handset from cradle; dial telephone number supplied by your system administrator.	2. Remove handset from cradle; dial telephone number supplied by your system administrator.
3. When you hear high-pitched tone, go on to Step 4. If line is busy or there is no answer, hang up and redial.	3. When you hear high-pitched tone, go on to Step 4. If line is busy or there is no answer, hang up and redial.
4. Press DATA button. DATA light should go on and keyboard unlock.	4. Place handset face down in coupler box; make sure cord is in slot. Close and latch coupler lid.
5. Place handset in cradle.	5. Turn on coupler switch within 20 seconds. Keyboard should unlock.
6. If DATA light goes out during session, retry from Step 1.	

Figure 7. Switched Line Terminal Connection Procedures

Conducting a 3767 (S/S) or 2741 Terminal Session

Once you have made the connection with the computer, you must contact VSPC by entering the VSPC logon command.

For the 2741 terminal, before you enter the logon command, it may be necessary to enter the following characters:

`/"nn`

where *nn* is a 2-digit number predefined by your system administrator. (Enter `/"nn` if an APL type element is installed.) Ask your system administrator if this entry is required for your 2741 terminal.

For the 3767 in start/stop mode and the 2741, you enter the logon command as follows:

```
vspc id=12345 @
ENTER PASSWORD
(you type the word TABOO, but it does not print legibly)
DATA 12:45:16 05/31/76 12345
YOU MAY BEGIN MAKING ENTRIES
READY
```

The @ (which represents the upshift of the 2 key) must be entered for the 2741 and for the 3767 operating in S/S mode.

Note: The VSPC command must be entered either entirely in capital letters or entirely in lowercase letters, as follows:

```
VSPC ID=12345
vspc id=12345
```

This is the one exception to the previously stated rules about command entry.

When you enter your logon password—TABOO—VSPC prints **XXXXXXXXXX**, over which you type the word TABOO. In this way, no one else can read the password you've typed in.

If you make a mistake when you are typing the password—suppose you typed TABU instead of TABOO—VSPC will prompt you again to enter the correct password. After you've made four attempts, VSPC will no longer allow you to continue. In this case, you must reenter the logon command.

After you've entered the correct password, VSPC prints the following information:

- Your profile *attribute* (DATA), the logon time (12:45:16), the date (05/31/76), and your user number (12345).
- An optional *logon message* (YOU MAY BEGIN MAKING ENTRIES) from the system administrator—this message can vary from one computer system to another; it can vary from day to day, and it sometimes may not be printed at all.

When you log on to VSPC using other terminal types (as, for example, the 3770 Data Communication System or the 3767 in SDLC mode), you enter the logon command as follows:

```
vspc id=12345
ENTER PASSWORD
(you type the word TABOO, but it does not display)
DATA 12:45:16 05/31/76 12345
YOU MAY BEGIN MAKING ENTRIES
READY
```

You'll note that there is only one difference in your logon command: you need not enter the upshift of the 2 key. (If you do enter it, VSPC accepts it, but does nothing with it.)

If you are using a 3270 terminal, press the CLEAR key before entering your logon command. If your logon is not accepted, reenter your logon and press the TEST REQ key instead of the ENTER key.

A terminal session is a series of interactions between VSPC and your terminal:

1. VSPC notifies you in one or more of the following ways that it is ready accept input:
 - Lights the ON LINE and PROCEED indicators.
 - Unlocks the keyboard.
 - Prints a message (for example, READY).
 - Prints a line number (for example, [10]).
2. You must then type a line of input as follows:
 - Type the line of information.
 - Press the carrier return (*cr*) key.
 - Wait for VSPC to respond before you type another line.

All of your VSPC processing follows this general form.

Note: Some users of 2741, 1050, and CPT-TWX terminals should not enter "99999" as the first five characters of a terminal input request; some users of 3270 terminals should not respond by pressing the test request key. Either of these actions may cause the terminal to be disconnected from VSPC. See your VSPC administrator to determine whether these restrictions apply to your organization.

Correcting and Deleting the Current Line

You can correct typing errors in your current line at any time before pressing *cr* using the following method:

1. Backspace to the error and signal attention.
2. VSPC responds with a series of actions:
 - Prints an underscore (`_`) beneath the erroneous character.
 - Backspaces one space.
 - Advances one line and unlocks the keyboard.
3. Type the remainder of the line correctly, press *cr*, and wait for the VSPC response.

You can delete the entire line you are currently typing as follows:

1. Signal attention without an immediately preceding backspace.
2. VSPC prints an asterisk (*), does a carrier return, and unlocks the keyboard.

Note: Some VSPC processors use a different method for correcting and deleting lines. See the appropriate terminal user's guide for information.

Using the Attention Key

In addition to character correction and line deletion, you can use the attention (ATTN) key for three other purposes:

- To signal attention. When your terminal is idle, pressing ATTN sends a signal to the system.
- To cancel output. When your terminal is printing, pressing ATTN ends the printing and sends a signal to the system.
- To escape from your current input request. When it is the only response to an input request, pressing ATTN lets you cancel that request.

The response to this signal varies depending on the foreground processor you are using.

Invalid Output Characters

If you find an N backspace Z (␣) character in your output, it is an indication that VSPC has found an invalid character in your data.

Conducting a 3270 Display Terminal Session

When you log on to VSPC using the 3270 Information Display System, you enter the VSPC (logon) command as follows:

```
vspc id=12345
```

Note: The VSPC command must be entered either entirely in capital letters or entirely in lowercase letters.

The VSPC Command Mode format appears, together with the message:

```
ENTER PASSWORD
```

You now type the word TABOO, but it does not display. After you've entered the correct password, VSPC displays the following information:

```
DATA 12:45:16 05/31/78 12345  
YOU MAY BEGIN MAKING ENTRIES  
READY
```

Which tells you:

- Your profile *attribute* (DATA), the logon time (12:45:16), the date (05/31/78), and your user number (12345).
- An optional *logon message* (YOU MAY BEGIN MAKING ENTRIES) from the system administrator—this message can vary from one computer system to another; it can vary from day to day, and it sometimes may not be printed at all.

If your logon is not accepted, reenter your logon and press the TEST REQ key instead of the ENTER key.

If your IBM 3277 or 3278 is connected via a 3791, the procedure is also different. Consult *VSPC Terminals* or check with your VSPC administrator.

If you make a mistake when you are typing the password—suppose you typed TABU instead of TABOO—VSPC will prompt you again to enter the correct password. After you've made four attempts, VSPC will no longer allow you to continue. In this case, you must reenter the logon command.

When you're using a 3270 display terminal, there are a number of VSPC terminal facilities available to you. You can use all of the terminal editing facilities (INS MODE, DEL, etc.) to modify the lines displayed in the input area, as described below.

Character Correction

Move the cursor back to the incorrect character with the cursor control key; then type the correct character.

Character Insertion

Press the INS MODE key and move the cursor to the position in which the additional character is to be inserted. Type the missing character. As characters are inserted, all characters to the right of the cursor are shifted to the right. If the insertion causes the line to exceed the line limit, the excess characters shift from the end of the first line of your input line to the beginning of the next. The INPUT INHIBITED indicator turns on and your keyboard is disabled if you try to insert more characters than the area will hold.

Press the RESET key to return the keyboard to its standard mode of operation and turn off the INSERT MODE indicator.

Character Deletion

Move the cursor to the position of the extra character and press the DEL key. The characters to the right of it move left one space.

Line Correction

With the backspace key, return the cursor to the first character of the data you want to replace. Then press the ERASE EOF key. This erases all characters from the cursor position to the end of the line. Retype the line.

Special Requests

Three terminal conditions allow you to make special requests.

1. When the MORE . . . PRESS CLEAR message is displayed on the screen, in addition to pressing CLEAR to continue output, you can:
 - Press PA1 to signal attention
 - Request backpaging
 - Request hardcopy

- Press PA2 to cancel output
 - Press ENTER to hold the display longer than the automatic change time of 30 seconds. (See the following section on Display Screen Hold.)
2. When ENTER INPUT appears on the screen, in addition to entering input, you can:
 - Press PA1 to escape from input
 - Request backpaging
 - Request hardcopy
 3. When the terminal is idle, you can:
 - Press PA1 to signal attention
 - Request backpaging
 - Request hardcopy

Display Screen Hold

When your output needs more than the remaining space available on the screen to display, VSPC pauses at the end of the screen with MORE . . . PRESS CLEAR shown in the screen separator line. Pressing CLEAR displays the next page. If, while MORE . . . PRESS CLEAR is showing, you press the ENTER key, the message changes to HELD . . . PRESS CLEAR and the current display remains until you press the CLEAR key.

If you press neither CLEAR nor ENTER within 30 seconds, the next page is displayed automatically.

Invalid Output Characters

If you find a double quotation mark (") in your output that you didn't expect, VSPC has found an invalid character in your data.

Using Program Function (PF) Keys – PFKEY Command

The program function keys provide shortcuts to entering commands, groups of characters, or control functions. Using the PFKEY command, you can assign a command, a group of characters, or a control function to any PF key on your terminal.

Presetting Character Groups

You can use the PFKEY command to preset a group of characters (that is, a *character-string*); the character-string is then inserted at the cursor position whenever you press the PF key. For example, if you enter the following command:

```
pfkey 5 'view *' noenter
READY
```

then during this terminal session, each time you press the PF5 key, the screen display is modified exactly as if you pressed the INS MODE key, keyed the characters VIEW *, and then pressed RESET. That is, the characters are placed at the location of the cursor, and the cursor and any characters following it are moved to follow the inserted character-string. This lets you use the

editing facilities, if you wish, to change the characters (for example, to change the * to a specific line number) before pressing ENTER and transmitting the result to VSPC.

Alternatively, you can specify:

```
pfkey 5 'view *' enter
```

Now, the characters are inserted in the same way, but VSPC proceeds as if you pressed the ENTER key immediately after pressing the PF5 key. In this case, if the cursor is in the command area when you press the PF5 key, you are issuing a VIEW * command.

When neither ENTER nor NOENTER is specified, ENTER is assumed.

Reentering the Last Input

You can use the PFKEY command to recall the last line entered and to place it at the cursor location. If you specify:

```
pfkey 4 last noenter
```

then during this session, whenever you press the PF4 key your last entered line is placed at the cursor position in exactly the same way as when you specify a character-string. You can use this function to recall and correct commands or input data that have been rejected for keying errors. You can then correct them using the terminal editing facilities and reenter them.

If you want to repeat lines *without* editing them, you can specify:

```
pfkey 4 last
```

This tells VSPC to reenter the last input line immediately after the PF4 key is pressed.

Moving the Screen Display

You can also use the PFKEY command to page backward or forward. For example, you specify:

```
pfkey 3 forward(2)  
READY
```

This lets you use the PF3 key, during this session, to page forward two screen pages each time you press it.

Similarly, you can use the PFKEY command to page backwards:

```
pfkey 2 backward(2)  
READY
```

This lets you use the PF2 key, during this session, to page backward two pages each time you press it.

Requesting Printed Output

If your display terminal has an associated printer, you can also set a PF key to request printing of the current page:

```
pfkey 1 hcpy  
READY
```

During this session, after you've issued this command, each time you press the PF1 key you're requesting a printed copy of your current page. (See also the following section on requesting printer services.)

Displaying PFKEY Settings

To get a display of the current PFKEY settings, enter the command PFKEY with no operands.

```
pfkey
```

Requesting Printer Services

If your 3270 display terminal has an associated printer, you can request that printer services be made available to you through the HARDCOPY command. For example, if you're using a 3275 display unit and you enter:

```
hardcopy *
```

you're requesting that the attached 3284 printer be connected for your use.

Later, when you want to print the current page, you can:

- position the cursor under the HCPY field in the control line and press ENTER, or
- select the HCPY field with the light pen, or
- press a PF key predefined as HCPY.

When you no longer need the services of the printer, you specify:

```
hardcopy off  
READY
```

which disconnects the printer from your terminal.

Note: If your 3270 Information Display System does not have a printer physically associated with your terminal, but has instead one that is shared by several terminal users, you can enter HARDCOPY *node-name* to request printer attachment. The *node-name* is a name specified by your organization to identify the shared printer.

Command Mode Entry and Editing

When you log on to VSPC, this form of entry and editing is in effect.

In Command Mode entry and editing, the screen format is that shown in Figure 8.

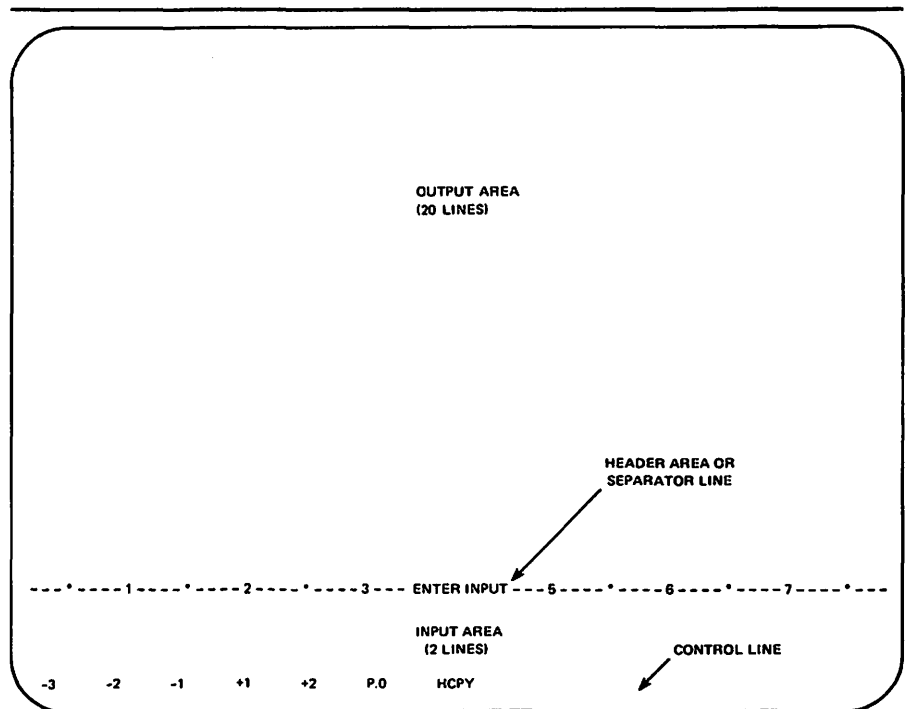


Figure 8. Command Mode 3270 Screen Display Format

The function of each area of the screen is:

- Control Line—in this area you specify terminal actions, either by positioning the cursor beneath a field and pressing ENTER, or by use of the optional light pen.
- Input Area—in this area you type commands or data for entry. In this area, you can use the terminal editing facilities (INS MODE, DEL, ERASE EOF, etc.) to edit the entry you're making before you press ENTER.
- Header Area—this area contains a display that tells you what kind of VSPC processing you're currently using.
- Output Area—when you type an entry in the input area and press ENTER, the entry disappears from the input area and appears in the output area.

When the entry is a command, the response to the command also appears in the output area, on the line following the command. This process continues until the entire 20 line area is filled.

When you're typing data lines, each time you press ENTER the current input line appears in the output area until the entire area is filled.

In either type of entry, when the entire screen is filled, VSPC saves it for your future reference and a new blank output area is displayed.

The general procedure for operating in Command Mode is:

1. VSPC notifies you in one or more of the following ways that it is ready to accept your input:
 - displays a question mark (?)
 - displays a message (for example, READY)
 - displays a line number (for example, 00010)
 - displays ENTER INPUT in the 3270 Header Area
2. You type a line of input as follows:
 - type the line of data or the command
 - press ENTER
 - wait for VSPC to respond in one or more of the above ways before you type another line

Note: Responses from VSPC can vary, depending on the type of terminal you're using and whether you're entering data, running a program, or entering commands.

Line-Numbered Data Entry

In Command Mode, you can enter data lines by typing the line number, an optional space, the data, and pressing ENTER (you can enter the lines in any numerical order). VSPC then positions the cursor so that you can enter another line. (For this example, we'll assume your workspace is empty.) For example, you type:

```
10 tabset 13 14 18 22 26 30
20 linesize 80
40 number * left 6
30 translate caps
50 enter data
```

However, VSPC treats the lines as if you'd specified them in the following order:

```
10 tabset 13 14 18 22 26 30
20 linesize 80
30 translate caps
40 number * left 6
50 enter data
```

By entering line numbers in increments of at least 10, you're able to enter new lines between already existing lines. For example, you decide to add one more statement:

```
15 name program1
```


VSPC places line 15 in numerical order between lines 10 and 20.

You can replace a line you've already entered simply by entering a new line with the same line number. For example, if you type:

```
10 tabset 10 20 30
```

This new line 10 completely replaces the original line 10 (10 TABSET 13 14 18 22 26 30), even though the new line is shorter than the original line 10.

Deleting Lines During Data Entry

To delete a line you've already entered, you enter the line number and then immediately press the ENTER key. For example, you want to delete line 00010 of the program you just wrote. You enter:

```
10 (you press ENTER)
```

Line 00010 now no longer exists in your workspace. VSPC does not tell you, however, that the line has been deleted.

Automatic Line Numbering – INPUT Command

When you are entering data, you can ask VSPC to generate line numbers for you, automatically. If your workspace is empty, you can request automatic line numbering as follows:

```
input  
00010
```

The VSPC response of 00010 indicates that this is your first line of data. (VSPC begins line numbering at 10 and increments each subsequent line by 10, as follows: 00010, 00020, 00030, etc.). VSPC displays a blank after each line number—you simply type in the data and press the ENTER key.

Note: You can specify INPUT PROMPT, and VSPC will respond in exactly the same way.

When your workspace is empty and you want the first line number to be 10, and subsequent numbers to be incremented by 5, you can specify:

```
input 10 5  
00010
```

Now VSPC will number the lines as follows: 00010, 00015, 00020, 00025, etc.

If you already have lines in your workspace (beginning at 10 and incrementing by 10, the last line being 70) automatic line numbering begins at the next line after the last existing one:

```
input  
00080
```

If you specify INPUT without any operands, VSPC automatically supplies you with the next higher multiple of 10 as a beginning line number with an automatic increment of 10.

If you want line numbers in multiples of 8, you can specify both the beginning line number, and the increment you want:

```
input 8 8
00008
```

(Now line numbering will be 8, 16, 24, 32, etc.)

Suppressing Line Number Printing – INPUT Command

You can also specify automatic line numbering without displaying the numbers at your terminal (this makes for quicker line entry), as follows:

```
input 40 8 noprompt
LINE 40
```

(VSPC positions the cursor at the beginning of the input area)

You can then continue entering lines of data. The succeeding line numbers (48, 56, etc.) will be generated in your workspace, but will not be printed during line entry. For example:

As Entered	In Workspace
LINE 56	
a=b	00056 A=B
x=y	00064 X=Y
c=a+x+y	00072 C=A+X+Y

Inserting Lines in a Workspace

When your workspace already contains data, and you enter the INPUT command without specifying any operands, VSPC begins generating line numbers so that you'll be adding data at the end of your workspace. You can also enter the INPUT command to begin inserting lines at some intermediate point in the workspace:

```
input 200
00210
```

You can now begin inserting lines at line 210. When you end automatic line numbering, by pressing ENTER before entering any data, VSPC changes all line numbers from line 210 on.

Note: Some foreground processors follow different rules about line numbering in this case.

Replacing Lines in a Workspace

The INPUT command also lets you replace existing lines in the workspace. For example, suppose that you want to replace lines 200 through 250 with new data. You can specify:

```
input 200 overlay
00200
```

OVERLAY tells VSPC to replace the existing line 200. Now you can type in the new data that replaces the old data. The next line number generated is 00210, which allows you to type in a new line 00210. You continue the process until you have replaced all the lines you want to replace.

Ending Automatic Line Numbering

When you've entered all the data you need and wish to begin entering commands, you press the ENTER key without entering any data as follows:

In PROMPT mode (that is, when line numbers are being printed)

00072 (you press ENTER)

READY

In NOPROMPT mode

(you press ENTER)

INPUT ENDED AT LINE 64

View Mode Entry and Editing

For display terminals, the View Mode entry and editing method lets you use all of the entry and editing methods you use in Command Mode. However, there are many additional facilities available for your use, as explained in the following paragraphs.

To change from Command to View Mode, you issue the VIEW command.

To change from View to Command Mode, press ENTER before you've made any changes to the current screen; alternatively, you can issue an AID, RUN, or STORE command, or an ENTER command that names an interpreter—all of which also change you back to Command Mode.

The View Mode entry/editing method uses the screen display format shown in Figure 9.

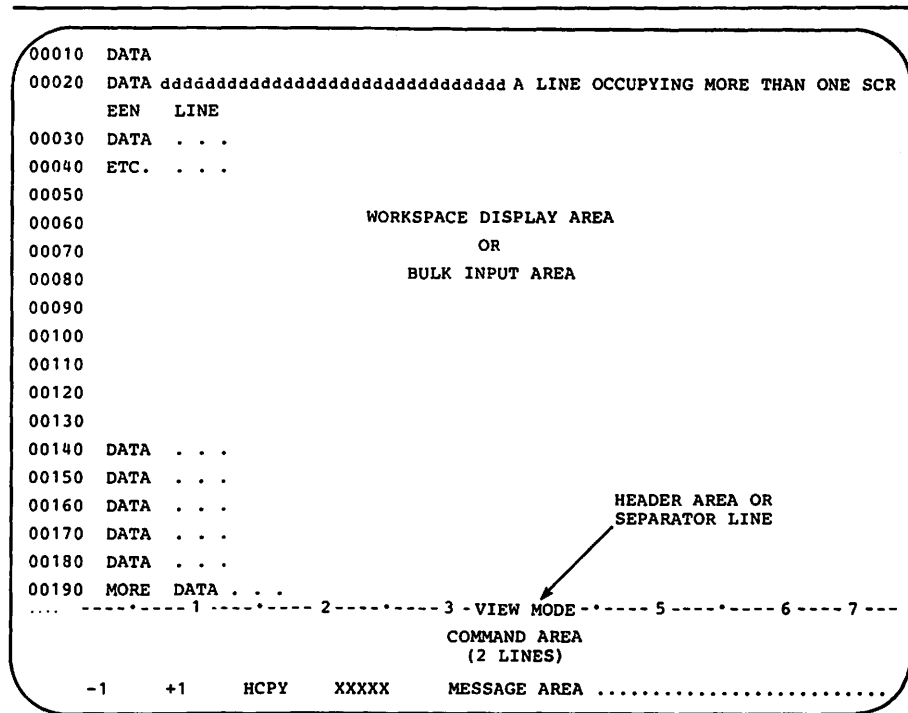


Figure 9. View Mode 3270 Screen Display

The function of each area shown is:

- Control Area—the left portion of the last line on the screen is the control area. In this area, you specify terminal actions, either by positioning the cursor under a field and pressing ENTER, or by use of the optional light pen.

- **Message Area**—the right portion of the last line on the screen displays VSPC responses to your commands and messages you receive.
- **Command Area**—this is the only area in which you can enter VSPC general commands. You can also use this area to type and edit data lines for entry into the workspace when you press the ENTER key. (For these uses, the command area corresponds to the input area in Command Mode.)
- **Header Area**—tells you the kind of processing (VIEW MODE or INPUT MODE) you're using.
- **Workspace Display/Bulk Input Area**—this area is the top 20 lines of the screen and is available for data entry or editing.

This area is the **Workspace Display Area** when you're in **VIEW MODE**—that is, in the mode that lets you edit existing files and enter data.

This area is the **Bulk Input Area** when you're in **INPUT MODE**—that is, in the mode that results after you enter an **INPUT** command.

In either mode, the current line is highlighted on your display screen.

Entering View Mode—VIEW Command

To enter View Mode, you issue:

```
view
```

and the screen format changes from that shown in Figure 8 to that shown in Figure 9. You can now begin using the View Mode facilities.

Changing the Screen Display—VIEW Command

You can use the **VIEW** command to move the screen display *without* changing the current line pointer. Thus, if you're already in View Mode, your current line is 00050, and you issue:

```
view 30
```

line 00030 becomes the first line displayed on the screen; however, the current line is still 00050.

Using the Control Area

You specify terminal actions by positioning the cursor under one of the following display areas and pressing ENTER, or by using the optional light pen:

```
-1      +1      HCPY      xxxxxx
```

-1 and **+1**—when selected, move the display backward (-1) or forward (+1) one screen across the workspace. (That is, the display is changed to show as many lines as will fit in the workspace display area. If +1 is pressed, the new display begins with the next line following the last line displayed on the current screen. If -1 is selected, the line preceding the top line of the current screen is the last line of the new display.) The current line pointer is unchanged.

HCPY—when selected, causes the current screen to be printed on the associated printer. (You must associate a printer by issuing a **HARDCOPY** command before you use this field.) The printed screen image includes the separator line, command area, and the control and message areas as well as the workspace display area.

XXXX—always displays the current line number. When selected, this field changes the display so that the current line is at the top of the screen.

(If you use the optional light pen to select fields and cause a change in the display screen, any changes you've made since you last pressed ENTER are lost. You can perform the same functions with the PFKEY command or by positioning the cursor and pressing ENTER, and such changes will be placed in the workspace and saved before the action is taken.)

Message Area

This portion of the screen displays messages you receive and VSPC responses to the commands you issue.

If a message or response is too long to fit in the message area, the display changes to Command Mode, MORE...PRESS CLEAR appears in the header area, and the message or response is displayed in the first 20 lines of the screen. Whenever you want, you can press CLEAR and the current screen is restored for your use.

Using the Workspace Display Area

The top 20 lines of the screen is your Workspace Display Area. Your current workspace is displayed, including its line numbers. You can move the cursor through the entire area and use the terminal editing keys (INS MODE, DEL, ERASE EOF, etc.) to add, replace, and delete text. If you change a line number, you are in effect moving that line. If the new line number is the same as one that already exists, the new line replaces the old one.

A line longer than 73 characters is displayed as multiple lines, as follows:

```
00050 THIS LINE CONTAINS MORE THAN 73 CHARACTERS  
      THIS LINE CONTAINS MORE THAN 73 CHARACTERS  
      THIS LINE CONTAINS MORE THAN 73 CHARACTERS
```

The blanks under the line number do *not* represent character positions in the data—they are simply blank *screen* characters that make the data easier to read. If you place data in them, however, the data is inserted in its correct line position. When you press ENTER, you'll discover that the line has been reformatted and the screen blanks have been reinserted under the line numbers.

Partial lines are never displayed. If the last few lines on the screen don't have space to display an entire multiple line, these screen lines are left blank and the line that doesn't fit on this screen will begin the next screen display.

View Mode cannot process more than 1000 characters in a line. If a line in a file is longer, it is not displayed and you cannot edit it. Instead, you'll get the message:

```
00290 LINE TOO LONG TO DISPLAY
```

at the point where the line should occur in the display. (You can use the SPLIT command, described later in this chapter, to break up the line so that you can display it.)

If you try to make additions to a line by modifying the line on the screen, so that it would become longer than 1000 characters, you'll get the following error message:

```
LINE TOO LONG
```

and the changes are not made.

If it's necessary, you can build such a line in parts and then later use the JOIN command to join the parts together. (The explanation on Using the JOIN Command, later in this chapter, tells you how to do this.)

Using the Bulk Input Area

When you're in View Mode and you enter the INPUT command, the top 20 lines of the screen become the Bulk Input Area, and the header area is changed to display INPUT MODE.

If you specify

```
input prompt
```

the screen is formatted with line numbers in the first five character positions in the line (as shown in Figure 9). If the workspace is empty, the line numbers begin at 00010 and are incremented by 10, unless you specify some other beginning line number and increment (see the section on Automatic Line Numbering—INPUT command).

If the workspace already contains data, and you enter INPUT PROMPT, the screen is formatted at the end of existing data, and line numbers are generated beginning at the next increment higher than the last existing line number.

You can also enter

```
input * prompt
```

and the screen input begins at the current line.

(See also the sections on Inserting Lines and Replacing Lines using the INPUT Command.)

If you don't want VSPC to display the line numbers, you can enter

```
input noprompt
```

and the first 20 lines of the screen are formatted as a blank area without any line numbers. You can now enter text anywhere in the 20 lines, using a substitute NEWLINE character to tell VSPC where one line ends and another begins, and where you want the line numbers inserted. (Whenever VSPC encounters a NEWLINE character, it ends that line, inserts a line number, and begins the next line.)

Whether you're using INPUT PROMPT or INPUT NOPROMPT, the blanks on the screen do *not* represent actual blank characters in the workspace; they are simply blank *screen* characters. Thus, if you want to insert blank characters in a line, you must actually enter them using the space bar or a tab.

After you've entered as much data on the screen as you want, or when the screen is full, press ENTER. This transmits the data from the terminal to your workspace and reformats the screen for more data entry.

If you're using INPUT PROMPT and the last line on the screen has not been used when you press ENTER, the system will transmit the data to your workspace and switch you out of INPUT mode.

Otherwise, when you want to switch out of INPUT mode, press ENTER before typing any data on the current screen.

Using Local Commands

When you're using View Mode, VSPC local commands are available to you; local commands affect only the immediate line(s) you're working with.

You specify a local command by typing it over the line number in the workspace display area. Local commands let you change the current line pointer, add lines, delete lines, and repeat lines. (In the examples that follow, the highlighted current line is shown in heavier type than the other lines.)

Changing the Current Line Pointer—/ Local Command

You can change the current line pointer by typing a slash over the line number. For example, you want to make line 00050 instead of 00060 the current line. You position the cursor under any digit of line 00050 and type the slash:

```
00040 data A  
00/50 data B  
00060 data C
```

Then, when you press ENTER, line 00050 becomes your current line:

```
00040 data A  
00050 data B  
00060 data C
```

(Note that the slash is replaced by the original zero after you press ENTER and the command is executed.)

You can enter the current line pointer local command more than once in one screen display; when you press ENTER, the current line is positioned at the last line you specified, and the preceding current line pointer commands are ignored. That is, in the previous example, if you enter a slash over *both* line 00040 and line 00050, line 00050 becomes the current line when ENTER is pressed.

Adding Lines—A Local Command

Using the A local command, you can add lines in the workspace. For example, if you want to add four *blank* lines after line 00050, you type 4A over 00050, as follows:

```
00040 data A  
4A050 data B  
00060 data C
```

When you press ENTER, the result is:

```
00040 data A  
00050 data B  
00051  
00052  
00053  
00054  
00060 data C
```

(Note that the current line is unchanged when the new lines are added.)

If you add eleven lines, then line 00060 and numbers following it are renumbered as far as necessary to maintain line sequencing.

Deleting Lines—D Local Command

You can use the D local command to delete lines at any point in the workspace. For example, you want to delete 4 lines beginning at line 00051. To do this you type 4D over line 00051, as follows:

```
00040 data A
00050 data B
04D51 data B1
00052 data B2
00053 data B3
00054 data B4
00060 data C
```

When you press ENTER, the result is:

```
00040 data A
00050 data B
00060 data C
```

Note that the current line has been deleted; the current line is now the line preceding the deleted lines. (You can also enter the D local command for a line that is *not* the current line; in this case, the current line is unchanged, unless it is within the range of deleted lines.)

Repeating Lines—R Local Command

Using the R local command, you can specify that a line should be repeated a given number of times. For example, to repeat 4 lines beginning at line 00050, you type 4R over the line 00050, as follows:

```
00040 data A
04R50 data B
00060 data C
```

When you press ENTER, the results are:

```
00040 data A
00050 data B
00051 data B
00052 data B
00053 data B
00054 data B
00060 data C
```

(Note that the current line is unchanged when the R local command is executed.)

Order of Processing

The order in which VSPC processes requests depends on the way in which you enter them:

1. If you use the light pen to select control-line fields, nothing else on the screen is processed.

For example, if you use the light pen to select HCPY, the copy printed is of your screen *before* you last pressed ENTER. The changes shown on the screen are entered into the workspace the next time you press the ENTER key.

2. If you use PF key defined with a PFKEY NOENTER command to set a character-string, nothing is changed in the workspace until you press ENTER.

If you use a PF key defined with a PFKEY ENTER command to set a character-string, it's as if you placed the character-string into the screen position of the cursor and then pressed the ENTER key.

3. If you request HCPY using a preset PFKEY or using cursor positioning followed by ENTER, changes on the screen are made in the workspace; then the screen is printed incorporating the changes.
4. When you press ENTER, changes you've made are placed in the workspace in the following order:
 - a. Line changes and D (delete) local commands, beginning at the top of the screen and continuing to the bottom.
 - b. The A (add) and R (repeat) local commands, beginning at the bottom of the screen and continuing to the top, and the bottom-most of any / (set current line pointer) commands.
 - c. Forward or backward paging requested by placing the cursor in the control line and pressing ENTER.
 - d. A PFKEY preset to FORWARD or BACKWARD.
 - e. General commands entered in the command area.

You can do the following at any time during the processing:

- If you press the PA1 key to signal attention, nothing else on the screen is processed.
- If you press the PA2 key while MORE...PRESS CLEAR is displayed, your output is canceled.
- If you press CLEAR, any screen changes you've made are canceled, and the screen is redisplayed in its original form.

Cursor Positioning

After processing the modifications to the screen, the new screen is formatted for the next display. When this new screen is displayed, the cursor is positioned at the start of the command area unless any of the following apply:

- You used a program function key with the NOENTER option to insert a set of characters. In this case the cursor is positioned after the string associated with the PF key.
- You issued the 'A' or 'R' local command. In this case the cursor is positioned at the start of the added lines. However, if another command entered on the same screen causes the workspace display to change such that the lines are not displayed, the cursor is at the start of the command area.

If you enter multiple 'A' and/or 'R' commands on the same screen, the cursor is set to the position corresponding to the command nearest the top of the screen.

- You issue the INPUT command. In this case the cursor is positioned at the start of the area for adding new text.

Session Oriented Commands

You can change the logon password with the **PASSWORD** command.

You can preset command operands using the **TEXT** command.

You can set logical tabs, backspaces, and newline characters through the **TABSET**, **BACKSPACE**, and **NEWLINE** commands.

Your terminal has features that let you set the physical line length and the physical tab settings. Through the **LINESIZE** and **TABSET** commands, **VSPC** lets you set the session line length and tab settings equal to the physical settings at the terminal. (If session settings and physical settings aren't the same, both input and output may be garbled and, perhaps, even unreadable.)

You can also specify whether or not you want your typed input data to be translated into capital letters through the **TRANSLATE** command.

Changing the Logon Password—PASSWORD Command

During a terminal session, you can change your current logon password. For example, you want to change your password from **TABOO** to **TOPSCRT**. By using the **PASSWORD** command, you can change it in one of two ways:

1. You can enter the complete command as follows:

```
password TOPSCRT  
READY
```

2. You can enter the command *without* specifying the password as follows:

```
password  
ENTER PASSWORD (you type the word TOPSCRT, but it does not display)  
READY
```

The next time you log on to **VSPC**, you will be required to furnish the password **TOPSCRT** instead of **TABOO**.

You can also use the **PASSWORD** command to remove your logon password by pressing **ENTER** when **VSPC** prompts you with the **ENTER PASSWORD** message.

Presetting Command Actions—TEXT Command

Certain **VSPC** commands—**LOCATE**, **CHANGE**, and **FIND**— have modifying operands that tell **VSPC** to respond either with a line number or with a line number and the text of the line.

- The **TEXT** operand specifies that the entire line is to be displayed.
- The **NOTEXT** operand specifies that only the line number of the affected line is to be displayed. (If both operands are omitted, this is the action ordinarily taken.)

The TEXT command lets you modify the action taken when the operand is omitted. For example, if you specify:

```
text on  
READY
```

you're telling VSPC to use TEXT when the operand is omitted in the LOCATE, CHANGE, and FIND commands. Thus, during this session, if you issue

```
locate 30  
00030 X + Y = Z
```

it operates as if you'd used an explicit TEXT operand.

Later in the session, if you specify

```
text off  
READY
```

and issue the same LOCATE command

```
locate 30  
00030
```

it operates as if you'd explicitly or implicitly specified NOTEXT in your LOCATE command.

The TEXT command works in much the same way for both the CHANGE and FIND commands. (See the section on Handling Workspace Data for descriptions of how these commands work.)

Setting Session Line Length—LINESIZE Command

When you log on to VSPC, the logical line length assumed for your terminal is as shown in Appendix A. You can change this default length with the LINESIZE command. Suppose you want to change the length to 66 characters. You specify:

```
linesize 66  
READY
```

The new session line length is now 66 characters. If you type an input line greater than 66 characters, VSPC accepts it just as you've typed it. When VSPC sends you an output line greater than 66 characters in length, the excess characters are displayed on the next line.

You must be sure the line length you specify is valid. For most terminals, it must be not less than 18 characters and not greater than 255. For example:

Valid LINESIZE Commands	Invalid LINESIZE Commands
linesize 30	linesize 10
linesize 79	linesize 256
linesize 255	linesize 1066

Note: While a line length of 255 is valid, it is greater in value than the physical line length of most terminals. If you specify such a line length, you may sometimes lose data that you're printing at your terminal.

Setting Session Tabs—TABSET Command

You should set the session tabs to match the terminal's physical tabs by using the TABSET command. Because VSPC doesn't recognize any tabs immediately after you log on, you must always issue a TABSET command before you tab any input data.

For example, your physical tabs are at the 4th, 8th, 12th, 16th, 20th, 24th, and 36th character positions on the line (counting the left margin as position one). You should therefore issue the following TABSET command:

```
tabset 4 8 12 16 20 24 36  
READY
```

The tab list you specify must have the numbers in ascending order; it can contain up to 26 tab positions.

Note: Except for terminals without physical tabs (like the 3270) and those for which VSPC automatically sets the tabs (3767 and 3770), the tabs you specify must match your terminal tabs. If they don't, both input and output will be different from what you intend.

Setting Logical Terminal Characters

VSPC has a logical character-setting feature that's useful in helping you format material such as printed reports of various kinds of data you're entering into your workspace.

The computer uses a set of nonprintable control characters to recognize when to insert a tab, when to backspace, when to begin a new line. By letting you equate a printable character to these nonprintable ones, VSPC lets you enter these control characters from terminals that do not have keys for these functions; you can also display the presence of these nonprintable control characters in any data you'll want to format for printing or for a formatted display.

The NEWLINE, BACKSPACE, and TABSET commands do this.

Using the NEWLINE Command

If you want to enter multiple VSPC commands on one line or enter newline characters into your workspace, you can specify:

```
newline 'ç'  
READY
```

Then, during this session, each time you type ç, VSPC translates it into the nonprintable control character that signals the beginning of a new line.

This feature is useful, for example, when you're entering commands. If ç is your NEWLINE character, you can enter:

```
tabset 13 ç linesize 78 ç translate caps ç enter xfortran
```

VSPC treats these lines as if you'd entered:

```
tabset 13  
linesize 78  
translate caps  
enter xfortran
```

If you're entering data using input mode, the substitute newline character tells VSPC where to begin each new line. In this case, when VSPC encounters a substitute newline character in your input, it ends the current line, generates a line number and a space, and begins the next line with the character following the substitute newline character.

To remove a substitute newline character, you specify:

```
newline ''  
READY
```

After this command is executed, the substitute newline character is no longer in effect.

Using the TABSET Command

With the TABSET command, you can define a substitute tab character, for typing convenience or to save storage space.

If you want to define a logical tab, you can specify

```
tabset 5 15 25 '%' expand  
READY
```

During this session tabs are set at line positions 5, 15, and 25; and VSPC interprets any input % to be a tab control character. Each time you enter a line with a % character in it, VSPC inserts the number of blanks needed to reach the next tab position you've specified for this session. Thus, if the % character represents the tab control character, when you enter:

```
00010 % translate caps
```

The data is represented in the workspace as follows:

```
00010      TRANSLATE CAPS
```

When TABSET EXPAND is in effect, you must be sure, when you type the substitute tab character, that your position on the current line is within the range of the tab positions you've set. That is, in the examples above, if you enter a % character when your cursor position is greater than 25, you'll get an error message telling you that you've attempted to enter an invalid character.

If you want to control tabbing later during printing or display, you can specify:

```
tabset 5 15 25 '%' noexpand  
READY
```

Then, during this session, tabs are set at line positions 5, 15, and 25; and VSPC translates any input % character into the nonprintable tab control character. However, VSPC does *not* insert any blanks into the workspace data. Thus, if you enter:

```
00010 % translate caps
```

The line is stored in your workspace as:

```
00010 t TRANSLATE CAPS
```

(where the "t" character shown represents the nonprintable tab control character actually stored).

Now, when you request the line to be printed or displayed, it will appear as:

```
00010      TRANSLATE CAPS
```

However, if you specify new tab settings through the **TABSET** command, the displayed output of lines containing tab characters will shift to conform to the new tab settings.

When **TABSET NOEXPAND** is in effect and you're displaying the workspace, any tab control character in a line beyond the line position of the last tab is displayed as an invalid character.

| To remove the definition of a substitute tab character, you specify:

```
tabset ''  
READY
```

| After this command is executed, there is no substitute tab character in effect.

Using the **BACKSPACE** Command

The **BACKSPACE** command lets you define a substitute backspace control character if your terminal (a 3270 Information Display, for example) does not have a backspace key. For example:

```
backspace '#'  
READY
```

Then, during this session, each time you type the # character, **VSPC** translates it into the nonprintable backspace control character. Thus, if you enter:

```
n#_e#_w#_
```

or

```
new###__
```

VSPC translates each # character into the nonprintable backspace control character when storing it in your workspace.

If you display this line at your 3270 terminal, **VSPC** retranslates the nonprintable backspace character into the substitute backspace character, and the display appears as:

```
N#_E#_W#_
```

or

```
NEW###__
```

because the 3270 display terminal does not have backspacing capabilities.

To remove a substitute backspace character, you specify:

```
backspace ''  
READY
```

After the command is executed, the substitute backspace character is no longer in effect.

Character Translation—TRANSLATE Command

Assuming you are not on a start/stop terminal with an **APL** type element, you can type your entries in either lowercase letters or capital letters. Immediately after you log on, **VSPC** translates every lowercase letter you type into a capital letter before placing it in your workspace. For example:

```
005 IF A = B  
005 If A = b  
005 if a = B  
005 if a = b
```

are accepted by VSPC as equivalent.

If you wish, however, you can specify that *alphabetic data* is not to be translated into capitals, but placed in your workspace exactly as you entered it. You do this by entering:

```
translate off  
READY
```

In every subsequent entry, any alphabetic data you enter will be placed in your workspace exactly as you entered it. In the preceding example, the four entries would no longer be equivalent.

If, later in your session, you do want VSPC to translate your entries, you type:

```
translate caps  
READY
```

and the original translation into capital letters is resumed.

Only alphabetic data is affected when TRANSLATE CAPS is specified. Commands are always translated into capitals, even if TRANSLATE OFF is in effect.

Within a command (ones that find or change data) alphabetic data characters within quotation marks may be included. Such data characters are used as input by the command—they are the characters to be found or changed. Therefore, these data characters *are* affected by the TRANSLATE command.

For example, if you want to locate occurrences of the capital letter sequence ABCDE in your workspace, and TRANSLATE CAPS is in effect, you can specify the FIND command in any of the following ways:

```
find 'ABCDE'  
find 'abcde'  
find 'AbCdE'  
find 'aBcDe'
```

and VSPC will search for ABCDE.

If TRANSLATE OFF is in effect and you want to find the same combination of characters, then you must specify:

```
find 'ABCDE'
```

because VSPC now does no translation of alphabetic data.

If you log on to VSPC on a start/stop terminal, using an APL type element, then VSPC will assume TRANSLATE APL and interpret input and output accordingly. (The '=' in VSPC ID=12345 is the upshift of the 5 key in APL. This non-standard position informs VSPC you will be using APL.) If you replace the APL element during your session, you should enter the command TRANSLATE CAPS or TRANSLATE OFF to inform VSPC of the change. Conversely, if you replace the standard element with an APL element, you should enter TRANSLATE APL in order that VSPC will interpret input and output as APL.

Note: If you are using a foreground processor, your system administrator may have defined a special translation convention you must specify before working with the processor. If this is necessary, your system administrator can tell you the special name you must use in the TRANSLATE command.

Checking VSPC Usage—QUERY Command

There may be times when you need information about your current terminal settings or about information that's in your user profile—if, for example, you get called away from the terminal and now that you're resuming work, you're unsure of your tab settings and translation convention.

The VSPC QUERY command lets you check for this information. If you specify:

```
query tabset  
4 8 12 16 20 24 36
```

VSPC prints the list of current tab settings for you. Then, if you specify:

```
query translate  
CAPS
```

VSPC tells you which TRANSLATE operand is in effect.

In VSPC, the most general form of query is to enter a command name with no operands. If that command sets parameters for your workspace or terminal session, the current settings will be displayed. For example, if you specify:

```
number
```

VSPC prints the number characteristics of the current workspace.

If you want to check on information in your user profile, you can specify:

```
query profile  
USERNUM: 12345  
LIBTYPE: PRIVATE  
CONTENT: DATA  
CPUMAX: 5 SECONDS  
SPACE: 2,500,000 BYTES  
SIZE: 40,000 EDITABLE, 80,000 OBJECT  
INTPRMAX: 60,000 MAXIMUM, 30,000 DEFAULT  
SSMAX: 4096 CHARS 10 ITEMS  
JECODE: DEPT87  
PROJLIB: 22222
```

(The meaning of each of these messages is given in the "QUERY Command—to Display User's Status" entry in the "VSPC General User's Dictionary" chapter.)

Using the QUERY command, you can also ask for accounting information (by specifying QUERY ACCOUNT) and workspace status information (by specifying QUERY WORKSPACE).

Changing Your Workspace Attribute—ENTER Command

At the time your user profile was established, your VSPC administrator specified the default attribute your workspace assumes (which is DATA for these examples). This is the attribute it has every time you log on.

You can explicitly change the attribute through the ENTER command. For example, if you want to change your attribute to CLIST, you type:

```
enter clist  
READY
```

Your workspace now has the CLIST attribute, and you are able to use your workspace as a command list. You can later, if you wish, use another ENTER command to change your workspace attribute back to DATA.

If your organization uses one of the IBM program product foreground processors, or a user-written processor, you can also specify that your workspace attribute is a *processor-name*.

You can specify the ENTER command any time that you must change your workspace attribute.

No matter what attribute you're using at the end of one session, your default attribute (which is DATA) will be in effect at the beginning of your next terminal session.

Note: The ENTER command *explicitly* changes your workspace attribute. The LOAD and RUN commands can *implicitly* change it; when the file named in these commands is placed in your workspace, the file's attribute replaces your present workspace attribute.

Saving Your Work Automatically

There can be times when your terminal session ends unexpectedly; there may be a communication line failure, a power failure, or the system administrator may be halting VSPC. When this happens, VSPC tries to save your current work for you in a special file named CONTINUE.

When a system administrator halts VSPC, VSPC tells you whether or not it was able to save your work, with ending messages like the following:

```
14:22:00 06/30/76 12345  
CONNECTED 01:30:30  
CPU TIME: 12.3
```

This series of messages means your work was saved. Depending on the options your organization has chosen, CONNECTED and/or CPU TIME may be replaced by a message like the following:

```
PROCESSING 3.01 UNITS
```

If VSPC couldn't save your work, you'll receive one additional message:

```
UNABLE TO SAVE 'CONTINUE'
```

VSPC may be able to save your work, even if you haven't received these messages (if, for example, a communication line failed and your terminal was disconnected). If VSPC saved your work, then the next time you log on to VSPC, the VSPC response could be:

```
DATA 16:22:22 06/03/76 12345
CONTINUE CLIST PASSWORD
YOU CAN NOW BEGIN PROCESSING
READY
```

The message **CONTINUE CLIST PASSWORD** tells you that your work has been saved, it has the attribute **CLIST**, and that you specified a password for it at some previous time. You can now retrieve the work from **CONTINUE** (supplying the correct password) and then go on working with it. (When you bring the work into your workspace (with the **LOAD** command), your attribute is implicitly changed to **CLIST**.)

Ending a Terminal Session—OFF Command

To end the terminal session, you enter:

```
off
14:22:22 06/05/76 12345
CONNECTED 01:30:10
CPU TIME 8.1 SEC
```

Here's what the VSPC ending messages mean:

- **14:22:22 06/05/76 12345** tells you the time and date of logoff and your user number.
- **CONNECTED** tells you the time elapsed since you logged on, in hours:minutes:seconds.
- **CPU TIME** tells you the amount of time, in seconds, you actually used the central processing unit.

CONNECTED and/or **CPU TIME** may be replaced by your organization as mentioned above.

- **PROCESSING UNITS** tells you how many accounting units you've used in processing your requests.

You can specify that you want your current work saved in the **CONTINUE** file, by typing:

```
off continue
```

VSPC gives you the same response—and, unless you receive an error message (**UNABLE TO SAVE 'CONTINUE'**)—you know that your work will be available the next time you log on.

Using VSPC Message Facilities

Usually during a terminal session you can send messages to other VSPC terminal users and (when you're not entering input or receiving output) receive messages from them as well. In addition, you can specify that you want to keep your terminal continuously ready to receive messages, or that you want to prevent messages from being routed to your terminal.

Sending Messages—SEND Command

To transmit messages to other users, you issue the SEND command. For example, if you want to send the message LET 'ER RIP to user 12350, you enter the following:

```
send 'LET 'ER RIP' 12350  
READY
```

If you want to send the same message to the computing system or VSPC operator, respectively, you use the keyword CONSOLE or OPERATOR in place of the user number (12350).

Note that the message is enclosed in single quotation marks (equivalent to the apostrophe). Therefore, we must have a method to distinguish an apostrophe within the message text from the one ending the message. In VSPC, two successive apostrophes within a message represent one text apostrophe character. Thus, the message actually prints as follows:

```
12345: LET 'ER RIP
```

Your user number, as the originator (or OPER if the operator is the originator), is printed first, and then the text is printed in capital letters, even though it was entered in lowercase letters. This is always true, even if you and/or the recipient has TRANSLATE OFF in effect.

If you send the message, but terminal user 12350 is not currently logged on to VSPC, or if you've made a mistake in the user number and there is no user number 12350, VSPC tells you so.

Receiving Messages—MESSAGE Command

Usually, your terminal can receive messages from other users at the following times:

- After a command has completed execution and before you've begun to enter another
- After you've completed entering a line of data and before you've begun entering another

If you don't want to receive messages from other users at all, you can then instruct VSPC not to route them to you, as follows:

```
message block  
READY
```

If later during the terminal session you want to be able to receive messages, you issue the following command:

```
message open
READY
```

As soon as the command is transmitted, you're able to receive messages from other users. If the VSPC operator sent a message to all users while you were blocking messages, you will immediately receive that message.

You can also (in case, for example, you're expecting important messages to be transmitted) specify that you want your terminal to be continuously open for message receipt. To do this, you specify:

```
message wait
READY
```

After you've executed this command, your terminal locks and you cannot enter either commands or data until you've signaled attention. Immediately after you've made an entry, your terminal keyboard locks again and again prevents further input until you've signaled attention. To return to normal VSPC operating mode, you signal attention and enter:

```
message open
READY
```

Identifying Messages

Every message—whether it is one you send or one that VSPC sends you—has a unique identifying header in the form:

```
ASUnnn
```

where *nnn* is a 3-digit number.

Ordinarily, the header is not printed with the message. However, there may be special circumstances when you need to know it. For example, you've entered a command and have received an error message that you want to look up in the message explanations listing (see Appendix B). In this case, you can enter:

```
message id
ASU201 READY
```

Now you can reenter the erroneous command, and the same error message is displayed, but this time it is preceded by the message header. For example (if user number 12350 is not logged on):

```
send 'LET 'ER RIP' 12350
ASU603 12350 NOT LOGGED ON
```

Later, if you no longer wish the headers to print, you can specify:

```
message noid
READY
```

An alternative method of receiving an explanation of an error message is to use the AID command "?". For example:

```
12350 NOT LOGGED ON
```

```
?
```

```
THE USER SPECIFIED IN YOUR COMMAND IS NOT LOGGED ON.  
THE COMMAND COULD NOT BE EXECUTED.
```

Handling Workspace Data

A number of VSPC general commands let you manipulate — that is, edit — the data you enter in your workspace. (For this reason, any data you enter is known as *editable data*.) The following sections describe the VSPC commands that edit data.

When you're using a 3270 display terminal, you always enter the general commands in lines 22 and 23 of the screen display.

The general command is executed when you press the ENTER key.

In the examples that follow, we'll assume that the workspace is numbered in increments of 10.

Erasing Data—DELETE Command

Previously, you've seen how to delete a single line at your terminal or in your workspace. The DELETE command allows you to erase one or more lines after they've been placed in your workspace.

For example, you want to erase line 190 from your workspace. You do it as follows:

```
delete 190  
1 LINES DELETED
```

When you delete a single line, the line number you specify must actually exist in the workspace.

You can erase a series of lines from your workspace by specifying a range of lines as follows:

```
delete 210:280  
8 LINES DELETED
```

You can also specify DELETE 206:284 (neither of which line numbers actually exist), and VSPC deletes all existing lines within the range specified—that is lines 210 through 280.

Rearranging Lines—MOVE Command

You can reposition lines in your workspace through the MOVE command.

For example, on inspecting the logic of the program in your workspace, you realize that line 280 should follow line 180 in sequence. (Remember that line 190 has already been deleted.) You can reposition the line using the MOVE command:

```
move 280 190
1 LINES MOVED
```

In this case, line 280 (the line to be moved) must actually exist. The destination line (190) does not actually exist in the workspace. After the MOVE command is executed, old line 280 has been repositioned at line 190 and has been renumbered as line 190. There is now no line 280 in the workspace.

On further inspection, you see that the processing done in lines 420 through 460 should logically take place after line 270 (line 280 has just been deleted), and so you specify another MOVE command:

```
move 420:460 280
LINES RENUMBERED FROM 290
5 LINES MOVED
```

Here you have specified that a range of numbers is to be moved (the colon (:) specifies that this is a range of numbers) and that these lines are to be placed beginning at line 280 and renumbered as lines 280, 290, 300, 310, and 320. Old line 290 now becomes line 330—and so VSPC sends you the message LINES RENUMBERED FROM 290 to tell you so. The second message tells you how many lines were moved to the new position.

To avoid renumbering of lines, you can specify a line increment by which the lines are to be renumbered, as follows:

```
move 420:460 280 2
5 LINES MOVED
```

In this example, the five lines have been repositioned beginning at line 280, and have been renumbered as 280, 282, 284, 286, and 288. Now the original line 290 retains its line number.

The destination line number can actually exist in the workspace, and VSPC will place the moved lines immediately following it, as follows:

```
move 210:250 410
5 LINES MOVED
```

In this case, the five lines are moved to a position *following* line 410, and are renumbered as 420, 430, 440, 450, and 460. Since these lines do not already exist, no renumbering is required.

Note: VSPC renumbers existing lines *only* if the move causes overlap of line numbers following the destination line.

In a MOVE command, the lines to be moved must be within the range of existing line numbers, though they need not actually exist. If a specified line number does not actually exist, VSPC assumes that the existing line numbers within the range you specify are the lines you mean.

Repetition of Data Lines—COPY Command

You can use the COPY command to duplicate one or more existing lines in another part of the workspace.

Copying a Single Line

You can duplicate one line, as follows:

```
copy 290 400
READY
```

Line 290 is copied when you press ENTER. Assuming that the line increment is 10 and that line 00400 does not actually exist, the duplicate line is given line number 00400; if line 00400 already exists, the duplicate line is assigned line number 00410 (or if another increment is in effect, that increment higher). If necessary, lines beyond 00410 are renumbered. If this happens, you get the following message:

```
LINES RENUMBERED FROM 410
```

Copying a Range of Lines

You can duplicate more than one line in the workspace by specifying a range of lines to be copied:

```
copy 290:310 400
READY
```

Lines 00290, 00300, and 00310 are then copied when you press ENTER (assume that the line increment is 10):

- If line 00400 does not exist, they're copied as 00400, 00410, 00420.
- If line 00400 does exist, they're copied as 00410, 00420, 00430; and lines are renumbered to maintain line sequencing.

You can copy one or more lines and make sure you avoid renumbering. For example, if the line increment is 10, you can copy a single line as follows:

```
copy 290 400 5
READY
```

When you press ENTER, line 00290 is copied at or after line 00400. The 5 tells VSPC to use this increment in calculating the new line number:

- If line 00400 does not exist, it's copied as line 00400.
- If line 00400 does exist, it's copied as line 00405.

Similarly, for a range of lines, you can specify:

```
copy 290:310 400 2
READY
```

- If line 00400 does not exist, they're copied as 00400, 00402, 00404.
- If line 00400 does exist, they're copied as 00402, 00404, 00406.
Renumbering will not take place, since the next line in the workspace is line 00410.

Reformatting Lines—JOIN and SPLIT Commands

The maximum number of characters in a workspace line entered from any terminal is 1000. (For a 3270 display terminal, the maximum that can be entered from or displayed at the terminal is usually much less.) There may be times when you need a line longer than the workspace or display maximum.

In this case, you can combine two consecutive lines into one, using the JOIN command.

For example, line 00290 in your workspace contains 650 characters and line 00300 contains 500. You can join these two lines together, by specifying:

```
join 290  
READY
```

Line 00300 has now been appended at the end of line 00290, and the next workspace line is now 00310. The resulting line 00290 is 1150 characters long, which means you can edit it at a 3270 display terminal only through Command Mode editing. If you display the workspace, line 00290 appears as:

```
00290 LINE TOO LONG
```

If you want to display or edit this line, you can take it apart again, using the SPLIT command:

```
split 290 500  
READY
```

After this SPLIT command is executed, the first 500 characters of old line 00290 are still contained in 00290; the remaining 650 characters are assigned a higher line number, as follows:

- If the next line number is 310 or higher, the new line is 00300.
- If the next line number is less than 00310, the new line number is half way between line number 00290 and the next line number. That is, if the next line number is 00302, the new line number is 00296.
- If the next line number is one higher, the new line is given this number and existing lines are renumbered in increments of 1 as far as necessary to maintain line sequencing.

Replacing Data—CHANGE Command

We previously showed you how to make corrections during line entry. You can also replace characters in workspace lines—in one line, in a range of lines, or throughout the entire workspace. The CHANGE command allows you to do this.

Changing Lines in Entire Workspace

You can change every occurrence of some character combination in your workspace. For example, in a processor program, you have decided to change the name A3 to the name B3 throughout the program, as follows:

```
change all 'A3' 'B3' text  
(the 33 changed lines are then displayed in numerical order)  
33 LINES CHANGED
```

By specifying TEXT, you are requesting that all the changed lines be displayed. When you're using View Mode, they won't fit in the message area: therefore, they're displayed in the workspace display or output area.

The message MORE . . . PRESS CLEAR is displayed on the header line. The changed lines in the workspace display area are also shown, as follows:

1. On the first display the first 20 changed lines are shown.
2. When you press CLEAR, the last 13 changed lines are shown, together with the message 33 LINES CHANGED.
3. When you press CLEAR again, the screen display of your workspace is restored.

Changing a Single Line

Suppose that line 120 of your source program contains the character string 'ANSWER: THE ANSWER IS', which you now want to replace with the character string 'ANSWER: THE QUOTIENT IS'. You specify the CHANGE command as follows:

```
change 120 'ANSWER ' 'QUOTIENT ' text
00120 PRINT 'ANSWER: THE QUOTIENT IS' B3
3 LINES CHANGED
```

Since you've specified only one line number, only that one line is changed.

Note that the space following ANSWER is included in the character string to be changed. This differentiates the second word ANSWER from the first in the character string. If we had specified 'ANSWER', the line would have been changed to:

```
00120 PRINT 'QUOTIENT: THE QUOTIENT IS' B3
```

because every occurrence of the character string ANSWER in line 120 would be changed.

Changing a Range of Lines

You now want to change B3 to C3 only for the range of lines 190 through 260. You specify, therefore, the range of lines you want changed:

```
change 190:260 'B3' 'C3' notext
190 200 240
3 LINES CHANGED
```

Because you specified NOTEXT, the changed lines are not printed, but their line numbers are.

If you do not specify TEXT or NOTEXT, the display is determined by the last TEXT command (see the section on Presetting Command Actions—TEXT command).

Deleting Characters from a Line

You can use the CHANGE command to delete characters from a line as follows:

```
change 120 ' ANSWER: ' text
00120 PRINT 'THE QUOTIENT IS' B3
1 LINES CHANGED
```

(By not specifying any replacement characters, you are erasing ANSWER: .)

Adding Characters to a Line

You can use the CHANGE command to add characters at the end of a line as follows:

```
change 120 '' ''CASES'' text
00120 PRINT 'THE QUOTIENT IS' B3 'CASES'
1 LINES CHANGED
```

Note that when you specify a single quotation mark as one of the characters to be added (or changed) you must specify two. Thus we have specified three pairs of quotation marks beginning and ending the new text. One pair is used by the CHANGE command; the other 2 pairs indicate one pair within the character string itself.

Note: In the CHANGE command, as in the MOVE command, the line numbers, when specified as a range, need not actually exist in the workspace. VSPC assumes that all existing line numbers within the specified range are the lines you mean.

Locating Data—FIND Command

You can use the FIND command to locate occurrences of character sequences in your workspace, in a way very similar to the CHANGE command.

Searching the Entire Workspace for Data

To search the entire workspace, you omit the range of line numbers. For example, to find every occurrence of the name B3 in a processor source program, you can specify the FIND command as follows:

```
find 'B3'
(a table of all the line numbers in which B3 occurs is displayed; the first found line is displayed at the top of the screen. The current line pointer is not changed.)
```

Searching a Range of Lines for Data

You can search only a portion of the workspace for occurrences of B3. If you want the search to begin at line 300 and continue to the end of the workspace, you specify:

```
find 300:99999 'B3'
(a table of line numbers in which B3 occurs. from line 300 on, is displayed)
```

300 specifies the beginning of the range; 99999 (the highest possible line number the workspace can have) indicates that you want to search all succeeding data.

In none of the previous examples have we asked that the found lines be displayed. Now we will search lines 100 through 150 for occurrences of B3 and ask that the found lines be printed:

```
find 100:150 'B3' text
00110 B3=A3/C3
00120 PRINT 'THE QUOTIENT IS' B3
```

Only if TEXT is in effect, are the lines displayed. (When TEXT is in effect, and you don't want to display all the lines, you can press PA2 or signal attention. Display immediately stops, although command execution is completed.)

When TEXT is omitted (or when NOTEXT is specified), only a table of line numbers is printed.

Note: In the FIND command, as in the MOVE command, the line numbers, when specified as a range, need not actually exist in the workspace. VSPC assumes that all existing line numbers within the specified range are the lines you mean.

Changing Line Numbering—RENUMBER Command

You can change the line numbers for all or part of your workspace with the RENUMBER command, and you can specify an increment to be used in the process.

Renumbering the Entire Workspace

Your workspace contains, for example, lines 10 through 510, with line 190 missing. To renumber the workspace with uniform increments of 10, you can specify:

```
renumber
32 LINES RENUMBERED
```

In your workspace, line 200 has been renumbered as line 190, and each succeeding line number has been similarly decremented. The workspace now contains lines 10 through 500.

Renumbering Part of the Workspace

Because you know that renumbering must begin after line 180, you can get the same result by specifying the first line number to be affected as follows:

```
renumber 200 190
32 LINES RENUMBERED
```

As in the previous example, line 200 is renumbered as line 190, and each succeeding line number is similarly decremented.

Always make sure, when you're specifying renumbering, that you're not requesting duplication of existing line numbers. For example, if you specify:

```
renumber 200 150
NEW LINE NUMBER WOULD CAUSE OVERLAP
```

the RENUMBER command is not executed, and VSPC sends you this message because the new line numbering would cause duplication of existing lines 150 through 180.

Changing the Line Number Increment

You can change the increment by which line numbers increase in value. Suppose you want to change a processor source program so that line numbering begins at 1 and increases by 1. You can do it as follows:

```
renumber 10 1 1
50 LINES RENUMBERED
```

Specifying 10 and 1 tells VSPC to change the initial line number from 10 to 1. Specifying the final 1 tells VSPC to increment the new line numbers by 1. The fifty lines of the source program now contain consecutive line numbers from 00001 to 00050.

Naming Your Workspace—NAME Command

You've entered your source program from the terminal, and now it's ready to be used—you've done all the editing you want to do. Now you can save it in a library. An earlier version has already been saved as QUOTIENT in your private library, and you want to replace that earlier version; therefore, you must use the NAME command to identify this new version.

The library you can place it in can be your own private library with libnum 12345; it can also be your project library with libnum 22222; or it can be any public library with libnum 100, 200, or 300. The naming convention you use depends on the library you want to save it in.

You're going to name your new version QUOTIENT.

To associate it with your own private library, you can omit the libnum:

```
name quotient
WARNING - FILE ALREADY EXISTS
```

Although VSPC sends you the warning message, the NAME command is executed, and your program now has the filename 12345 QUOTIENT.

(The warning message tells you that there is already a file in your private library that has the name QUOTIENT.)

To associate your program with your project library, you can specify the filename as follows:

```
name 22222 quotient
READY
```

Your program now has the filename 22222 QUOTIENT.

Alternatively, to associate your program with your project library, you can specify:

```
name 0 quotient
READY
```

The program again has the filename 22222 QUOTIENT.

To associate your program with public library 200, you can specify:

```
name 200 quotient
READY
```

Your program would now have the filename 200 QUOTIENT.

Saving Workspace Contents—SAVE Command

Having named your workspace, you can now place it in a library as follows:

```
save  
READY
```

The workspace contents have now been placed in the library you chose under one of the following filenames:

Library	Filename
private library	12345 QUOTIENT
project library	22222 QUOTIENT
public library 200	200 QUOTIENT

In your private library, because you're replacing an earlier version with a new one, you must specify the **NAME** command before the **SAVE** command is issued. However, for the project and public libraries, you can execute the **SAVE** command and name the workspace at the same time. For example:

```
save 200 quotient  
READY
```

The workspace contents have now been filed in public library 200, and the workspace has the name **QUOTIENT**.

You can also use the **SAVE** command to change the name of the workspace, and save it in your private library as follows:

```
save mathprob  
READY
```

The new source program now exists in your private library as **12345 MATHPROB**.

Note: After the **SAVE** command is successfully executed, the workspace still contains the data you've had copied into the library.

Displaying Your Workspace—LIST Command

You can, if you want, display your program at the terminal by specifying:

```
list
```

and every line of data in your workspace is displayed at your terminal. On the display screen, as many lines as will fit on the current screen are displayed, and **MORE . . . PRESS CLEAR** is shown on the header line. When you press **CLEAR**, the next screenful is displayed, and **MORE . . . PRESS CLEAR** is again shown if all the requested lines have not displayed. This cycle continues until all the requested lines have been displayed.

You may wish to check only part of your workspace contents. For example:

LIST Command	What is Displayed
<code>list 200</code>	line 200
<code>list 100:200</code>	lines 100 through 200 inclusive
<code>list 10:200</code>	lines 10 through 200 inclusive
<code>list *</code>	current line

The first example displays line 200.

The second example displays a section from the middle of your workspace.

The third example displays your workspace from the beginning to an intermediate point.

The fourth example displays the current line.

If you don't know the exact line-number at which you wish to begin the display, VSPC will take the next higher line number as the one you are looking for.

For example, you've inserted lines 214 and 216 between lines 210 and 220, but you can't remember the exact line numbers you used. If you specify:

```
list 211:99999
```

The display will begin at line 214 and continue to the end of your data.

Erasing an Entire Workspace—CLEAR Command

You've made all the corrections you want on QUOTIENT/QUIET, you've saved it, and now you want to go on to other processing. To provide an empty workspace for further and different work, you use the CLEAR command as follows:

```
clear  
READY
```

The workspace now has no name or password, contains nothing, and any library characteristics (explained in the next section) are removed. The current workspace attribute is, however, retained. That is, if your profile attribute is DATA, but you have changed it to CLIST, then after the CLEAR command is executed, your current workspace attribute is still CLIST.

Erasing a Partial Workspace—EXTRACT Command

You can erase all but a specified line from your workspace, as follows:

```
extract 190  
READY
```

Your workspace now contains one line with line number 00190 and has no name or password. However, the current workspace attribute and library characteristics (explained in the next section) are retained.

You can also erase all but a specified range of lines, as follows:

```
extract 20:60  
READY
```

Your workspace now contains five lines with line numbers 00020 through 00060. The current line pointer is set as follows:

- If the current line was lower than any retained line, the current line pointer is set to the first retained line.
- If the current line was within the range of the retained lines, the current line pointer is unchanged.
- If the current line was higher than any retained line, the current line pointer is set to the last retained line.

Using VSPC Libraries

The VSPC library commands control access to the VSPC libraries available to you. There are commands for placing a file into your workspace, for creating and removing files, for specifying access control, and for obtaining a directory of current filenames. Each is explained in the following paragraphs.

Obtaining a File—LOAD Command

You can make your file named 200 QUOTIENT available for further corrections by using the LOAD command, as follows:

```
load 200 quotient
READY
```

After the LOAD command is executed, the QUOTIENT program has been placed in your workspace (which now also has the name QUOTIENT), ready for any further processing you may wish to do.

If your attribute, at the time the LOAD command is executed, is CLIST, and the QUOTIENT attribute is BASIC, when the LOAD command is executed, the workspace attribute is changed to BASIC.

Combining Files—MERGE Command

You may want to combine one file with another. You can do this using the MERGE command.

For example, your private library contains one file of data on male students (named MFILE) at McKinley Junior High School and a separate file on female students (named FFILE) at the same school. You can use the MERGE command to combine the two files. First you load the FFILE:

```
load ffile
READY
```

FFILE now is present in your workspace. (Note that because FFILE is from your own library, you can omit the library number when specifying the LOAD command.) You now can combine FFILE with MFILE by using the MERGE command:

```
merge mfile
READY
```

The two files have now been combined in your workspace. All of the entries for female students are first in the file, and then all of the entries for male students.

You can now use the combined file for immediate processing, store it under another filename, and so forth.

File Insertion—MERGE Command

You can also use the MERGE command to insert one file at a specific position within another file. For example, you have written a processor program (named BALANCE) that calculates and prints a bank balance for one customer at a time. You want to insert into it, after statement 100, an

all-purpose date-conversion routine (named 100 DATEDATA) that can be used with any such processor program. The line number increment for the inserted lines is to be 5. You can use the LOAD and MERGE commands to combine the programs:

```
load balance
READY
merge 100 datedata 100 5
RENUMBERED FROM LINE 110
READY
```

The DATEDATA routine now begins at line 105 of the BALANCE program; old line 110 of the BALANCE program now follows the last line of the DATEDATA routine, and has a line number five higher.

File Interleaving—MERGE Command

Using the MERGE command, you can also intersperse records from one file between records of another file. Suppose you have a data file of employees (named PERSFILE), the line numbers of which represent the employee identification numbers. You have built a new file of new employees (named CHGFILE), that also includes updated information on current employees. Now you want to combine the two files in ascending order according to line (identification) number. First you can load PERSFILE into your workspace. Then you can use the following form of the MERGE command:

```
merge chgfile overlay
READY
```

Based on the ascending order of line numbers (that is, the employee identification numbers), the two files are now combined into one.

Figure 10 shows how records from the files are combined.

Old PERSFILE	CHGFILE	New PERSFILE (after MERGE execution)
10100...	10105...	10100...
10110...	10111...	10105...
10115 NONX...	10115 EX...	10110...
		10111...
		10115 EX...

Figure 10. Using MERGE with OVERLAY

Creating and Modifying Files—FILE Command

You can use the FILE command to create directory entries for new library members and to modify directory entries for existing files.

Creating Files—FILE Command

You want to create a directory entry for the new file CHGFILE, and you don't want the file to exceed 10,000 characters in length at any one time. The file will be a VSPC sequential file. You can enter the FILE command as follows:

```
file chgfile 10 sequential
READY
```

10 specifies the maximum size for the file as 10,000 bytes (characters). You can specify or omit SEQUENTIAL when you wish the file to be a sequential file.

The directory entry has now been written; however, the actual space for the file will not be allocated until actual data is put into it (through, for example, the SAVE command).

You can also specify that the file will be a VSPC direct file (in this case, all records in the file must be of the same length, and line numbers must start with 1 and increment by 1). If CHGFILE is to have these characteristics, you can specify the FILE command as follows:

```
file chgfile 10 direct
READY
```

The directory entry in both of the previous examples has the file attribute DATA. If you wanted to create a file with a different attribute, you could specify the CONTENT operand. If you did not specify the file limit size, your user profile limit for editable workspaces would be used.

Modifying Files—FILE Command

You can also use the FILE command to modify the size and/or the characteristic of the file.

Suppose that you originally specified CHGFILE as a direct file, with maximum size of 10,000 characters. You now want to modify that specification by reducing the maximum file size and changing the file type to sequential. (The current size of the file is 9,556 characters.) You specify the FILE command as follows:

```
file chgfile 8 sequential
WARNING - FILE LIMIT LESS THAN FILE SIZE
```

The new specification has been entered into the directory; however, the current contents of the file have *not* been changed. The warning message simply informs you that this is *not* a new file, and that when you modify this file you must reduce the file length to 8,000 characters or less, or you will not be able to save it.

If the current contents were 7,332 characters, then you would receive the following message:

```
WARNING - FILE ALREADY EXISTS
```

and the new specification again would take effect.

Assigning Sequence Numbers—NUMBER Command

Through the NUMBER command, you can associate a sequence number format with a file that you own. You can specify the NUMBER command for the contents of the workspace or for sequential or direct files in the VSPC library.

The sequence numbering you specify is used when 80-character card image records are produced by the SUBMIT command (see “Submitting Batch Jobs—SUBMIT Command”). The sequence number is generated from the VSPC line number for the record.

You can specify that sequence numbers are to be placed at either the beginning or the ending of the 80-character record. You can also specify that no sequence numbers are to be generated.

For file JCL1, you want sequence numbers in the last eight positions of the card image; thus you specify:

```
number jcl1 right 8  
READY
```

or

```
number jcl1 right  
READY
```

If the number of character positions is omitted for RIGHT, 8 is assumed.

For file COB1, you want sequence numbers in the leftmost six positions of the card image; thus you specify:

```
number cob1 left 6  
READY
```

or

```
number cob1 left  
READY
```

If the number of character positions is omitted for LEFT, 6 is assumed.

You can specify one through eight character positions for either LEFT or RIGHT sequence numbering.

For file DAT2, you want to omit sequence numbers in the card image; thus, you specify:

```
number dat2 none  
READY
```

You can also specify a sequence number format for the lines contained in your workspace. For example, you could place several files in your workspace and specify a sequence number format for them through the following series of commands:

```
load jcl1  
READY  
merge jcl2  
READY  
merge lnk1  
READY  
merge jcl3  
READY  
merge dat2  
READY
```

```
number * right
READY
```

Note: The * specifies that the NUMBER command applies to the lines in the workspace.

Now when you issue a SUBMIT command for the workspace, the sequence numbers will appear in the rightmost eight character positions of each card image record. When the SUBMIT command is executed, VSPC ensures that the sequence number plus the text plus any blanks used for padding result in an 80-character record. If any generated sequence number or any record is too long to fit in the space allotted, VSPC sends you an error message, and the SUBMIT command is not executed.

Note: The NUMBER command sequence numbering convention is used by the SUBMIT command (see the “Processing Batch Jobs” section), and also by the VSPC Service Program control statements (which you can use with the assistance of your system administrator).

Restricting File Access—PROTECT Command

The PROTECT command allows you to specify (or remove) a password for a file, and to restrict other users’ access to that file.

Specifying a Password—PROTECT Command

Suppose you want to apply a password to 100 DATEDATA. You can specify the PROTECT command as follows:

```
protect 100 datedata password
ENTER NEW PASSWORD FOR 100 DATEDATA
(you type the word SCRT, but it does not display)
READY
```

Once this PROTECT command is executed, you must supply the password “SCRT” before you can access 100 DATEDATA.

Preventing File Modification—PROTECT Command

You can use the PROTECT command to prevent your own accidental modification of a file as follows:

```
protect 100 datedata/scrt nowrite
READY
```

Once this PROTECT command is executed, you, the owner, can obtain 100 DATEDATA and read it, but you cannot purge the file or make any changes to it.

Restricting File Reading—PROTECT Command

You can use the PROTECT command to allow other users only to use 100 DATEDATA as a source program without discovering its contents, as follows:

```
protect 100 datedata/scrt noread
READY
```

After this PROTECT command is executed, other users can load 100 DATEDATA for processing, but they cannot successfully execute a SAVE, STORE, CHANGE, FIND, or LIST command.

Combining PROTECT Command Operands

You can combine any of the operands in one specification of the PROTECT command:

```
protect 100 datedata/scrt password noread nowrite
ENTER NEW PASSWORD FOR 100 DATEDATA
(you type the word NIX, but it does not display)
READY
```

Once this PROTECT command is executed, a user must supply password "NIX" to access the file. After loading the file, other users cannot read it; they can only use it as a source program for processing. You, the owner, are unable to modify it.

Removing PROTECT Command Restrictions

You can also use the PROTECT command to remove restrictions you have previously placed on a file. For example:

```
protect 100 datedata/nix nopass read
```

After this PROTECT command is executed, a user can access 100 datedata without supplying a password and also read its contents. You are still unable to change the contents of the file (NOWRITE is still in effect).

Extending File Access—SHARE Command

You can make any file you own available to all VSPC users by specifying the SHARE command. You can also withdraw the permission to access a file.

For example, BALANCE is a file in your private library with PROTECT NOWRITE in effect. You can make it available to all VSPC users by specifying:

```
share balance yes
READY
```

or

```
share balance
READY
```

Note that this command does *not* change the previous NOWRITE specification for this file. You are still unable to modify it. (If NOREAD were in effect, other users would be able to use BALANCE for processing, as a source program, but would not be able to discover its contents.)

Whether you specify or omit the YES operand, the SHARE command has the same meaning—the file is made available to other users.

Now you want to withdraw permission to access **BALANCE**. You specify:

```
share balance no  
READY
```

After execution, the previous **SHARE YES** specification has been removed, and other users can no longer access **BALANCE**.

If **BALANCE** is a file in library 22222 (your project library), then you can specify the **SHARE** command as follows:

```
share 22222 balance  
READY
```

or

```
share 0 balance yes  
READY
```

Users who do not belong to your project library are now able to access **BALANCE**. Users who do belong to your project library can already access **BALANCE**.

Note that the **SHARE** command has no meaning when issued for a public file.

Transferring Files – ACQUIRE and RELEASE Commands

If you own files in a noncontrolled project library (and you aren't the library manager), you can make any file you own available for transfer of ownership to another project library member. You do this by specifying, for example:

```
release 22222 quotient  
READY
```

When this **RELEASE** command is executed, file 22222 **QUOTIENT** is made *available* for another user of noncontrolled project library 22222 to acquire ownership; however, you still own the file.

After you have executed the **RELEASE** command, another user or you yourself (but not the library manager) can issue an **ACQUIRE** command:

```
acquire 22222 quotient  
READY
```

If another project library member executes this **ACQUIRE** command, the ownership of the file is transferred to that library member, and the space it occupies is subtracted from your library space and added to the other library member's space.

If you yourself issue this **ACQUIRE** command, you're retracting your previous **RELEASE** command; after the **ACQUIRE** command is executed the file is no longer eligible for ownership transfer.

Note: The noncontrolled project library manager can also issue the **RELEASE** command, but only for files that he does not own; the library manager cannot issue the **ACQUIRE** command to take over ownership of other users' project library files.

Displaying Library Directories—QUERY LIBRARY Command

You can obtain a list of the files in specific libraries—your own, your project library, or any public library.

To obtain a list of all files in your own library, you specify:

```
query library
(VSPC displays the names of all files in your own library)
```

To obtain a list of all command list files in your library, you specify:

```
query library content(clist)
(VSPC displays the names of all command list files in your library)
```

You can obtain such lists of your project library or any public library. For example, you want a list of all sequential files in public library 100. To obtain the directory, you must supply the public library number. Thus, you specify:

```
query library(100) type(sequential)
(VSPC displays the names of all sequential files in public library 100)
```

There are many more variations of the QUERY LIBRARY command possible. The “QUERY Command—to Display Library Directory” entry in the “VSPC General User’s Dictionary” chapter gives the possible combinations.

Displaying File Information—QUERY FILE Command

Another form of the QUERY command allows you to obtain detailed information about a specific filename in a library.

Suppose you want to obtain such information about BALANCE, which resides in your own private library. You specify the QUERY FILE command as follows:

```
query file(balance)
```

VSPC displays the following information:

```
FILE TYPE: SEQUENTIAL  CONTENT: DATA
FILE SIZE: 7556  LIMIT: 8000
NUMBER OF RECORDS: 121
SEQUENCE NUMBERS: RIGHT 8
ACCESS: NOWRITE
DATE LAST WRITTEN: 06/30/76
DATE LAST READ: 08/15/76
```

(FILE SIZE is the present file size, in characters; LIMIT is the maximum size allowed; SEQUENCE NUMBERS: RIGHT 8 says that sequence numbers generated by the SUBMIT command will be placed at the right margin and are 8 digits long. The other messages are self-explanatory.)

If BALANCE resided in library 22222 (your project library), or in a public library—and neither library is a controlled library—then another message would precede all the others:

```
OWNER: 12345
```

Removing Files—PURGE Command

You can use a PURGE command to delete a file from your own library. Suppose, for example, you want to remove BALANCE from your library. You specify the PURGE command as follows:

```
purge balance
FILE PROTECTED (NOWRITE)
```

VSPC tells you that you cannot purge BALANCE until you remove the NOWRITE protection characteristic. Thus you reenter the following series of commands:

```
protect balance write
READY
purge balance
READY
```

With the PROTECT command you change the NOWRITE characteristic to WRITE. Then you issue the PURGE command, and BALANCE is removed from your library.

Note: You can successfully execute the PURGE command *without* supplying the password for the file. If you do supply the password, however, it must be the correct one.

Using Foreground Processors

A foreground processor is a computer program that runs interactively under VSPC control. Many of them allow you to write problem solutions in easy-to-understand language and then translate your solutions into computer machine instructions for execution.

Certain IBM foreground processors can be used with VSPC. User-written foreground processors can also be designed to run under VSPC.

Any such foreground processor must be installed with VSPC before you can use it. Check with your system administrator whether any are available to you.

Two of the kinds of foreground processors you might use are interpreters and compilers – both explained in the following sections.

Interpreters

An interpreter acts as a middleman between you and VSPC (and the computer). When you're using an interpreter, it interacts with you directly and tells both VSPC and the computer what you want to do; after the actions have been completed, the interpreter tells you the results.

The IBM program product VS APL is an interpreter designed to be used with VSPC.

APL allows you to enter statements one at a time. Each statement is executed as soon as it is entered into the workspace.

In addition, you can write statements in a series that will be translated and executed as a group; such a group of statements is known as a function. A workspace containing APL functions can be saved for execution at any future time.

Immediately executable statements and functions can be intermixed in any sequence you please.

Once you enter APL, you are under complete control of the VS APL interpreter—which issues all necessary VSPC commands and computer machine instructions for you. See *VS APL for VSPC: Terminal User's Guide* for additional information.

Compilers

A compiler is a processor that allows you to write a whole series of problem-solving statements—known as a *source program*. The source program is then compiled—that is, translated into machine language—into an *object program*. The object program can then be executed to provide you with the solution to your problem.

When you use a compiler, there are always three distinct steps taken at three different times:

1. *Source Creation Time*—the time at which you enter source statements at the terminal. The source statements are considered as data—and are always preceded by a line number. The complete group of source statements is the *source program*.
2. *Compile Time*—the time at which the compiler processor translates the source program into machine language. During the process, the compiler program may find errors in your source program; in this case you will receive error messages telling you what you have done wrong. Some errors are so serious that the compilation cannot be completed; others are minor enough to allow a complete object program to be produced. The result of compilation—if serious errors have not been found—is a machine executable *object program*.
3. *Object Time*—the time at which the object program, using any necessary data you have provided, actually produces the problem solutions you want. Once compilation is complete, object program execution can follow immediately; alternatively, the object program can be saved for execution at some later time.

Compilers designed to be used with VSPC commands are VS BASIC, VSPC FORTRAN, and VSPC PL/I. See *VS BASIC for VSPC: Terminal User's Guide*, *VSPC FORTRAN Terminal User's Guide*, and *VSPC PL/I Terminal User's Guide* for additional information about using these program products.

User-written compilers can also be designed for operation with VSPC.

VSPC gives you facilities to compile and execute source programs, to execute object programs, and to store both source and object programs in VSPC libraries. The following sections explain how to use VSPC commands in conjunction with a compiler.

Setting Processor-name Attribute—ENTER Command

In order to use one of the compilers, you must set your workspace attribute to the proper *processor-name*, if such a processor-name is not your profile attribute.

If your profile attribute is not a compiler, you use the ENTER command to specify the desired processor-name. (See Figure 16 for a list of IBM program product processor-names.)

Writing the Source Program

Whatever compiler you are using, you must write the source program following the correct syntax and grammar rules for the language processor you are using or the processor will reject the program. Each line is entered as data—that is, each source line must be preceded by a VSPC line number. You can either type in the line numbers yourself or use the automatic line numbering provided by the INPUT command.

Once you have written the entire source program you can begin processing it.

Placing the Source in a Library—SAVE Command

For future reference, you may want to place the untranslated source program in a library. You do this through the SAVE command.

If there is no other file with the same name as your source program, you can name it and save it by using the SAVE *filename* form of the command:

```
save scephob  
READY
```

If you loaded an earlier version of the program (which has the same name) and updated it, you can also use the SAVE command without specifying a filename.

Remember that even after saving the source in the library, the contents of your workspace are unchanged.

(See the previous sections on “Naming your Workspace—NAME Command” and “Saving Workspace Contents—SAVE Command”.)

Compiling and Saving Your Program—STORE Command

You can compile your source program and place the machine language object program in your library. The object program is then available for execution at any future time.

You can use the STORE command to compile SCEPROB into a machine language object program, and to name the object program and place it in a library:

```
store objprob  
OBJPROB  
(a series of error messages from the compiler follow)  
COMPILED PROGRAM NOT STORED  
TIME 1.1 SECS
```

You thought your source program was error free. However, during the translation process, the compiler processor found syntax and/or semantic errors in it. Thus, VSPC sends you an explanatory message (COMPILED PROGRAM NOT STORED). The series of compiler error messages explain what is wrong.

You can now edit the source program to correct the indicated errors, save it again, and again try to store it:

```
store objprob
OBJPROB 06/29/76 14:22:22
COMPILED SIZE 8192 BYTES
TIME 1.5 SECS
```

The above series of messages tells you that your source program was error free (insofar as the compiler was able to tell), that the size of the object program is 8192 bytes (characters) in length, and that it took the central processing unit 1.5 seconds to compile it. Your machine language object program is now placed in your library, and is available under the name OBJPROB; your source program is still available in the workspace.

Note: The TIME message may be replaced by a message like the following:

```
PROCESSING: 0.1 UNITS
```

Executing Your Program—RUN Command

You've saved copies of both the source program and the object program in your library. Now you want to execute it.

If your source program is a one-time problem solution, you will probably only want to execute the program once. There is no need, therefore, for you to save either the source program or the object program for future reference.

In this case, you can use the RUN command to compile and execute the source program as it exists in your workspace:

```
run
SCEPROB 06/29/76 14:33:00
(lines of SCEPROB execution output)
TIME 5.6 SECS
```

When this form of the RUN command is executed, the workspace contents are compiled and executed. At the end of execution, your SCEPROB still exists in the workspace.

You can also compile and execute a source program you have saved in a library. You specify the filename in your RUN command:

```
run scephob
SCEPROB 06/29/76 14:33:00
(lines of SCEPROB execution output)
TIME 5.6 SECS
```

The RUN command with a filename has the effect of a LOAD filename command, followed by a RUN command. Therefore, in the above case, the workspace is cleared and the SCEPROB source program is loaded into the workspace; then the program is compiled and executed. At the end of execution, the workspace still contains the source program that was loaded as part of the RUN command processing.

You can use the RUN command to execute an object program from your library. You've already used the STORE command to compile and save the object program (as explained in the preceding section); now you issue the following RUN command:

```
run objprob
OBJPROB 06/29/76 14:33:00
(lines of OBJPROB execution output)
TIME 4.1 SECS
```

When this form of the RUN command is specified, the contents of the workspace are erased and the object program is executed. At the end of execution, your workspace is empty and unnamed.

Assigning Unit Numbers—ALLOCATE Command

If you're using the IBM VSPC FORTRAN program product—or a similar user-written compiler that has file assignment capabilities—you can use the ALLOCATE command to assign your terminal or a VSPC file to a unit number. (The VSPC unit number is equivalent to the dataset-identifier used in input/output statements in the FORTRAN language.)

A VSPC foreground processor can also use unit number assignments to identify external VSAM files in executing object programs or interpreter functions.

For example, if your source program uses unit number 16 for an input file and unit number 18 for an output file, then before you enter a RUN command, you can enter the following ALLOCATE commands.

For a FORTRAN input file (one you're going to read):

```
allocate * 16 session
READY
```

Your terminal input is now available to the object program as unit number 16. SESSION tells VSPC that this file assignment is to remain in effect until you log off.

For a FORTRAN output file (one you're going to write):

```
allocate chgfile 18 mod
READY
```

CHGFILE is now available to the object program as unit number 18. MOD specifies that data written by the object program will be added after any existing data in CHGFILE. (If MOD is omitted, the present contents of CHGFILE are replaced by new data and the object program creates a new file.) You omitted SESSION; therefore this file assignment remains in effect only during execution of this object program; you could also specify RUN and get the same effect.

Releasing Unit Numbers—FREE Command

The FREE command removes previous unit number allocations. You can use the FREE command to remove only one unit number allocation, as follows:

```
free 16
READY
```

The previous assignment of unit number 16 has now been removed.

You can also use the FREE command to remove all unit number allocations you've made:

```
free  
READY
```

Every ALLOCATE command you've entered during this session has now been removed.

Processing Command Lists

A command list is a series of VSPC commands entered into your workspace as data. VSPC commands are immediately executable, but a command list is not—it is first created and then executed later as a group of commands.

Once a command list is created it can be executed immediately, and/or it can be saved in a library for future reference. The sequence of commands can be executed using the RUN command.

The following sections explain how to create, save, and execute command lists.

Command List Creation

When you create a command list, you enter the series of commands, preceding each one with a line number. You can either type the line numbers yourself, or you can use INPUT mode to generate them automatically.

Once the entire list is completed, you must make sure the workspace attribute is CLIST before you process the list. Then you can execute the list immediately using the RUN command, and/or you can place it in a library using the SAVE command.

You can create command lists to execute a series of commands you know you'll always use whenever you're doing a particular type of processing.

For example, you could execute the following command list whenever you're going to create fixed format FORTRAN programs:

```
00010 tabset 13  
00020 linesize 78  
00030 translate caps  
00040 enter xfortran
```

Or you could execute the following command list whenever you're going to create a COBOL program to be compiled through the job entry facility:

```
00010 tabset 13 14 18 22 26 30 34 38 42 46 50 54  
00020 linesize 80  
00030 translate caps  
00040 number * left 6  
00050 enter data
```

In either case, you'll be able to use one RUN command in place of the series of commands in the list.

Figure 11 shows the series of actions necessary to create a more elaborate command list.

```
clear
READY
input
00010   protect persfile write
00020   load persfile
00030   merge chgfile overlay
00040   list line
00050   protect * password nowrite
00060   save
00070   purge chgfile
00080   file chgfile/ 8
00090   press ENTER
READY
enter clist
READY
save perlist/
ENTER NEW PASSWORD FOR PERSLIST
(you type VERBOTEN, but it does not display)
READY
```

Figure 11. Creating the PERSLIST Command List

The command list is used monthly to update a personnel file (PERSFILE) by merging in new and changed employee records (maintained in CHGFILE) that have accumulated for this month. PERSFILE is ordinarily protected against accidental updating (NOWRITE).

The sequence of commands specifies the following actions:

- Line 00010 removes the NOWRITE protection characteristic so that the updating process can take place.
- Line 00020 places PERSFILE in the workspace.
- Line 00030 places the changed records in numerical order from CHGFILE into PERSFILE.
- Line 00040 displays the new PERSFILE for checking.
- Line 00050 replaces the usual protection characteristics for PERSFILE.
- Line 00060 places the new version of PERSFILE in the library.
- Lines 00070 and 00080 remove the old version of CHGFILE from storage, and provide for a new version of CHGFILE to be added during the next month.

Once you've created the command list, you give the workspace the CLIST attribute. Then you save the PERSLIST in your library with the password VERBOTEN.

Command List Execution—*RUN Command*

Once you have created the PERSLIST command list, you can execute it from the workspace with a RUN command as follows:

```
run
(command list execution output follows)
```

Alternatively, after the PERSLIST has been placed in a library, you can use the RUN command to place it in the workspace and then execute it as follows:

```
run perslist
ENTER PASSWORD FOR PERSLIST
(you type VERBOTEN, but it does not display)
(command list execution output follows)
```

In either case, the execution output is as shown in Figure 12. VSPC prints each command in turn, executes it, and also prints any resulting messages. When terminal input is needed, the keyboard unlocks, and input is requested (lines 20, 30, 50, and 80 for password prompting). When the entire list has been executed, VSPC unlocks the keyboard for additional terminal entries.

```
* 10 PROTECT PERSFILE WRITE
READY
* 20 LOAD PERSFILE
ENTER PASSWORD FOR PERSFILE
(you type TABU, but it does not display)
READY
* 30 MERGE CHGFILE OVERLAY
ENTER PASSWORD FOR CHGFILE
(you type SCRT, but it does not display)
READY
* 40 LIST LINE
(the new version of PERSFILE is displayed)
* 50 PROTECT * PASSWORD NOWRITE
ENTER NEW PASSWORD FOR *
(you type LOCKUP, but it does not display)
READY
* 60 SAVE
READY
* 70 PURGE CHGFILE
READY
* 80 FILE CHGFILE/ 8
ENTER NEW PASSWORD FOR CHGFILE
(you type PASSNO, but it does not display)
READY
TIME 0.1 SECS
(keyboard unlocks for further input)
```

Figure 12. Executing the PERSLIST Command List

You can also execute a command list *without* displaying the commands themselves. In the preceding example, if you specify:

```
protect perslist noread
ENTER PASSWORD
(you type VERBOTEN, but it does not display)
READY
```

then, when you execute the command list, the output is similar to that shown in Figure 12, except that the command lines are not displayed. The VSPC responses to the commands (except for the READY messages) are printed and also the requests for password entry.

Command List Nesting—RUN Command

You can write command lists in such a way that one list will cause another list to be executed; when the second list completes execution, control is automatically returned to the first command list, beginning at the next command after the RUN command.

You use the RUN command within the first command list to begin the processing of the second. The second command list can itself contain a RUN command that causes the processing of a third command list. In this way, you can nest any multiple command lists.

For example, you could add three more commands to the PERSLIST command list, as follows:

```
00090 run libprint
00100 clear
00110 enter data
```

During PERSLIST execution, when

```
00090 RUN LIBPRINT
```

is executed, the LIBPRINT command list is brought into your workspace and executed.

When LIBPRINT execution is completed, the remaining commands of the PERSFILE command list are executed.

The TIME 0.1 SECONDS message is then displayed.

If any chained command list ends abnormally, then all of the command lists to which it's chained also end abnormally.

Processing Batch Jobs

Your organization may include VSPC remote job entry capabilities—that is, VSPC job entry commands that you can use to submit requests for batch job processing.

Through the VSPC job entry commands you can enter batch jobs from your terminal. You can specify the batch job input as coming from your workspace or from one or more files. Once the batch job is entered into the system, you can—from your terminal—request job status information, cancel the job if

necessary, erase job output if you wish, or request that the job output be made available at your terminal or printed on a high speed printer.

The remote job entry facilities are not always available for your use. All of the following must be true before you can submit batch jobs:

1. Your VSPC system must be initialized to allow remote job entry.
2. VSPC remote job entry capabilities must be currently available to users.
3. Your VSPC user profile must specify a job entry code you can use in submitting batch jobs. (If it does not, you're not allowed to use any of the remote job entry facilities.)

If all these conditions are met, you are able to submit batch jobs from the terminal, using the VSPC job control commands.

Note: If you attempt to use the job entry facilities and they are not currently available to you, you'll receive an error message.

Programming Considerations for Job Entry

When you submit batch jobs through VSPC, you must submit them either from your workspace, or as one or more VSPC files. These files must contain all of the instructions needed for the system to execute the job; the files must be included in the correct order.

The file(s) must contain valid job control language (JCL) for the system you're using and can also contain (when needed) source statements and data.

All three can be entered as data from the terminal and either submitted directly from the workspace or saved in a VSPC library for later submission.

By supplying the proper JCL statements, you can:

- compile, link-edit, and execute compiler source programs
- store and retrieve object programs and data in system libraries
- execute system utilities

In fact, all of the batch processing capabilities of the system are available for your use.

You can, for example, develop and test FORTRAN or BASIC programs using the VSPC foreground processors, and then use the VSPC job entry facilities for production runs of the fully tested program.

Suppose you want to develop a production program that you'll execute many times—an all-purpose program, for example, that computes deflection of I-bars given different dimensions and different materials—and you want to write the program in FORTRAN. You can develop and test the program interactively, using the VSPC FORTRAN foreground processor.

Once the I-bar program is executing correctly, you can use the VSPC job entry facilities to recompile the program as a batch job, and then save the object program for use whenever you want to execute the program later. Each time you want to execute the program, you can request batch execution from your terminal, and also request that the job output be made available at your terminal.

You can also use the job entry facilities to develop source programs, and to compile and execute them. Suppose, for example, you want to develop programs using COBOL, a programming language supported by OS/VS but not by VSPC. First, you create the needed VSPC files, as shown in Figure 13.

Each file can be saved in your VSPC library. Each then can be used by the VSPC remote job entry facilities.

VSPC submits each line of text as an 80-character record, and, if you request it, places a sequence number at the left or right margin. (See "Assigning Sequence Numbers—NUMBER Command".)

Each file can, of course, be saved with a password. In the following examples, we'll assume that passwords are omitted.

File	Contents
JCL1	JCL to invoke a COBOL compiler and to save object code in the proper system library
JCL2	JCL to link-edit object modules into load module libraries
JCL3	JCL for execution
LNK1	linkage-editor control cards to link-edit a single object module into a load module
LNK2	linkage-editor control cards to link-edit several object modules into one load module
COB1	a COBOL source program
COB2	a COBOL source program
COB3	a COBOL source program
DAT1	data to be processed by the COBOL programs
DAT2	data to be processed by the COBOL programs

Figure 13. Developing COBOL Batch Programs

Submitting Batch Jobs—SUBMIT Command

You use the SUBMIT command to request VSPC to enter a batch job stream into the system.

You can enter all the necessary JCL, source statements, and data in your workspace, and then type:

```
submit
JOB SUBMITTED AS C1234501
(or JOB SUBMITTED AS C1234501 0001)
READY
```

The first message tells you that the workspace was submitted as a job stream, and that VSPC has assigned the *jobname* C1234501 (or C1234501 0001) to it. (The VSPC-assigned name is made up of your job entry code (C12345), a VSPC-assigned number from 00 through 99 and, in OS/VS2 only, a JES2- or JES3-assigned number from 0000 through 9999.)

The second message tells you that you can now enter more input.

In the following examples, we'll use the filenames as illustrated in Figure 13 and jobnames that might be assigned under DOS/VS or OS/VS1.

You can also use the **SUBMIT** command to group one or more VSPC files as an input job stream. Suppose you want to compile a COBOL source program and save the object program, you can use the **SUBMIT filename** form of the command as follows:

```
submit jcl1 cob1
JOB SUBMITTED AS C1234502
READY
```

VSPC submits the two files as one job stream. They are placed in the stream in the order you specify them in the **SUBMIT** command. The jobname assigned has a different suffix than the previous one.

You can compile, link-edit, and execute a COBOL source program using the same form of the **SUBMIT** command:

```
submit jcl1 cob2 jcl2 lnk1 jcl3 dat1
JOB SUBMITTED AS C1234503
READY
```

Or you can link-edit and execute a COBOL object module already stored in a system library:

```
submit jcl2 lnk1 jcl3 dat2
JOB SUBMITTED AS C1234504
READY
```

In the **SUBMIT filename** form of the command, you can specify 1 through 15 filenames.

Interrupting SUBMIT Command Processing

You can signal attention to interrupt **SUBMIT** command processing at any time. Any jobs not yet processed are removed from the job stream. VSPC sends you the following message:

```
SUBMIT TERMINATED - 6 JOBS SUBMITTED
```

where 6 represents the number of jobs entered for processing before the attention interrupt.

Querying Job Execution—STATUS Command

You can instruct VSPC to query the system about the status of one previously-submitted job or of all your previously submitted jobs. To query the status of a single job, you can either use the entire VSPC-assigned jobname as the operand, or you can omit the job entry code.

For example, for job C1234501, you can specify:

```
status C1234501
or
status 01
(status message returned)
```

The status message returned can be any one of the following messages:

```
C1234501 AWAITING EXECUTION
C1234501 EXECUTING
C1234501 COMPLETED
C1234501 HELD
C1234501 NOT FOUND
```

You can also query the system about the status of all your previously-submitted jobs:

```
status
(status messages returned)
```

The status messages are the same as those previously listed with two exceptions:

- The JOB NOT FOUND message never appears
- A NO JOBS FOUND message is displayed if you have no jobs with VSPC-assigned names in the system

Obtaining Output at Terminal—LOAD Command

You can request that batch job output be placed in your workspace. You can request output from one job at a time, and you must use the VSPC-assigned jobname as an identifying operand.

Once the output is placed in your workspace, you can use VSPC commands—such as LIST, CHANGE, NAME, SAVE, etc.—to print it at your terminal, edit it, identify it, and place it in a library. (If the output is larger than your workspace, only the first portion that fits within your workspace is available.)

As explained in the following sections, execution results of the LOAD Job Entry Command depend on the computer system you are using.

| OS/VS—Obtaining Specific Data Sets

When your computer system is OS/VS1 or OS/VS2, you can request that output from a specific data set (or a range of logical records from a specific output data set) be made available:

```
LOAD OUTPUT( jobname ) DS(dataset-identifier )
```

or

```
LOAD OUTPUT( jobname ) DS(dataset-identifier )LINE(begin-line :end-line )
```

In OS/VS1, *dataset-identifier* is the number of the DD statement for the SYSOUT data set, counting only the SYSOUT=VSPC DD statements as they appear in the job control language. In OS/VS2, *dataset-identifier* is that part of the dataset-name that uniquely defines the data set—in JES2, the last four digits of the key-number in the last qualifier; in JES3, the ddname, stepname.ddname, or procstepname.stepname.ddname. In OS/VS2, *dataset-identifier* may be enclosed in single quotation marks within the parentheses, and must be so enclosed if *dataset-identifier* includes periods.

To determine the *dataset-identifier*, you request that system messages be made available by substituting SMSG for DS(*dataset-identifier*). (You must, in your original JCL JOB statements, specify MSGCLASS= *vspc-output-class*, where “*vspc-output-class*” is determined by your organization. You must also specify MSGLEVEL=1.) You type the LOAD command as follows:

```
load output (01) smsg
READY
```

You can then display the termination messages to determine the *dataset-identifier* you want to retrieve.

You have now determined that the dataset-identifier you want is 16, and that you want only lines through 200 placed in your workspace. Thus, you specify:

```
load output (01) ds(16) line(1:200)
READY
```

After the command is executed, lines 1 through 200 are available in your workspace.

Alternate Output Destinations—ROUTE Command

You can request that batch job output be sent to an alternative destination—a high-speed printer or a remote workstation, for example. You can request only one job at a time, and you must use the VSPC-assigned jobname as the identifying operand.

| OS/VS—Routing Entire Job Output

When you want to send the entire batch job output to an alternative destination, you specify:

```
route 01
READY
```

When the command is successfully executed, job C1234501 is routed to a default class defined by your system administrator.

You can also specify an explicit output class and/or remote workstation as follows:

```
route 01 class(6) dest(plum)
READY
```

or

```
route 01 class(6)
READY
```

or

```
route 01 dest(plum)
READY
```

When successfully executed, these forms of the ROUTE command send output to an explicit class and/or destination. CLASS(6) and DEST(PLUM) are merely examples; valid class numbers and destination identifiers are defined by your system administrator.

OS/VS—Routing Output Data Sets

When your computer system is OS/VS1 or OS/VS2, you can also request that output from a specific data set be sent to an alternative destination.

You determine the data set to specify by using the LOAD command (see "OS/VS—Obtaining Specific Data Sets"). Assuming that data set 16 is the one you need, you specify:

```
route 01 ds(16)
READY
```

When the command is successfully executed, data set 16 is routed to a class defined by your system administrator.

You can also explicitly specify an output class and/or remote workstation as follows:

```
route 01 class(6) dest(plum) ds(16)
READY
```

or

```
route 01 class(6) ds(16)
READY
```

or

```
route 01 dest(plum) ds(16)
READY
```

When successfully executed, these forms of the ROUTE command send data set 16 output to an explicit class and/or destination. CLASS(6) and DEST(PLUM) are merely examples; valid class numbers and destination identifiers are defined by your system administrator.

Canceling a Batch Job—CANCEL Command

You can request that a previously submitted batch job be canceled by the system. You identify the job by its VSPC-assigned jobname as follows:

```
cancel 01
READY
```

READY tells you that the command was successfully executed, and you can now enter more input.

If you enter a CANCEL command and execution is already complete, you'll get the following message:

```
cancel 01
JOB NOT CANCELED, OUTPUT ON QUEUE
```

Execution of job C1234501 is stopped by this form of the CANCEL command, but any job output already created is not erased. To erase it, you'll have to issue a SCRATCH command.

OS/VS—PURGE Operand

In OS/VS, you can request that any existing job output be erased—whether the job is currently executing or execution is complete—when you issue the CANCEL command, as follows:

```
cancel 01 purge  
READY
```

In OS/VS1, any existing job output is erased; however, any system messages are not erased.

In OS/VS2, both job output and system messages are erased.

OS/VS1 Only—Interrupting CANCEL Execution

In OS/VS1, you may issue a CANCEL command and get back the following message:

```
cancel 01  
BUSY - COMMAND BEING RETRIED
```

This message means that the system is internally attempting to resubmit your CANCEL command. You can wait for completion, but you will not be able to enter any input from the terminal until the CANCEL command is successfully executed.

Therefore, you may wish to withdraw your CANCEL request. To do this, you signal attention. VSPC responds with the following message:

```
CANCEL TERMINATED  
READY
```

You now can enter input from the terminal.

Erasing Job Output—SCRATCH Command

The SCRATCH command lets you erase job output. The form of the SCRATCH command you issue depends on the system you're using.

| OS/VS—Erasing Entire Job Output

You can erase all of your job output, as follows:

```
scratch 01  
READY
```

| After the command is executed, the entire job output is erased.

OS/VS—Erasing an Output Data Set

In OS/VS1 or OS/VS2, you can request that a specific data set be erased:

```
scratch 01 ds(16)
```

where 16 is an example of a dataset-identifier. The dataset-identifier to specify is decided in the same way as explained for the LOAD command.

Getting Help in Forming Commands—AID Commands

The AID commands let you receive online prompting in the formation of valid VSPC commands. You may want to change the word “maximal” to the word “major” wherever it appears in lines 9234 through 10911 of your workspace, but you have forgotten exactly how to specify the operands for the CHANGE command. You may use the command “?CHANGE”. VSPC AID replies with a message similar to the following:

```
LINES TO CHANGE?  
-ONE ( LINE )  
-RANGE ( LINE1 : LINE2 )  
( LINE NUM(S) OR NULL FOR ALL ):
```

If you wanted to change all the lines in your workspace where “maximal” might appear, you would reply by entering no character. Since you want to change only the lines within the range 9234-10911, you enter:

```
9234:10911
```

Typical prompts from VSPC AID, and your replies, continue something like the following:

```
CHARS TO REPLACE OR DELETE?  
( NULL TO ADD AT END OF LINE ): maximal  
NEW CHARS?  
( NULL FOR NONE ): major  
OUTPUT?  
1. CHANGED LINES  
2. LINE NUMBERS ONLY  
( 1,2 ):
```

Here VSPC AID is asking you to choose option 1 or 2. You reply:

```
1
```

VSPC then replies with a message telling you:

- Your change command is complete
- A summary of what you are changing

VSPC finishes with a question similar to:

```
EXECUTE?  
( Y,N ):
```

VSPC AID has prompted you to form the valid CHANGE command that it has displayed. You may now choose whether or not to execute the command by entering “Y” (for “yes”) or “N” (for “no”). Following your reply you will be returned to normal VSPC operation, and the CHANGE command will be executed, or not, as you chose.

You can also use the AID command “?” to receive an explanation of a message sent by VSPC. For example, if you had not understood the prompt:

**EXECUTE?
(Y , N) :**

you could have entered the command “?”. VSPC AID would have replied with a message telling you that:

- **The displayed command is ready to execute**
- **You can type N to leave AID without executing the command**
- **You can type Y or simply press ENTER to execute the command**
- **Predefined values will be used for omitted operands (this reminder is displayed because you may have skipped some of the prompting sequence by replying with an attention signal).**

See “General Execution Considerations” under “AID Commands.”

VSPC GENERAL USER'S DICTIONARY

VSPC general commands and related VSPC topics are defined in the following pages. (VSPC supervisory commands are described in *VS Personal Computing (VSPC) Installation Reference Material.*) For easy reference, the entries are listed in alphabetic order. An overall description of VSPC command functions is given in Figure 14.

VSPC topics that relate to a number of commands are described in separate entries.

Entries for VSPC commands are described in the following order:

Brief Definition: a generalized description of the actions the command performs.

Format: shows the valid combinations of operands and any restrictions.

Operand Descriptions: define the meaning of each operand. Rules for the formation of user-defined portions of operands are given, as well as any restrictions on operands or portions of operands.

General Execution Considerations: describe actions taken when the command is executed, as well as information about interactions between commands. The VSPC response to successful execution is usually the READY message. When the VSPC response to successful execution results in some other message, that response is given.

Command	Action
?	Asks for AID explanation or prompting
ACquire	Takes ownership of noncontrolled project library file
ALlocate	Assigns a file a processor unit-number
Backspace	Assigns logical backspace character
CAnceL	Cancels a previously submitted job
CHange	Alters a group of data characters
CLear	Erases a workspace
COpy	Duplicates workspace lines
Delete	Erases lines from a workspace
ENter	Sets the workspace attribute
EXtract	Deletes all but a specified range of lines from a workspace
FILE	Creates or modifies a file
FINd	Locates a group of data characters
FRee	Removes previous unit-number assignments
Hardcopy	Allocates a 3270 attached printer
Input	Begins automatic VSPC line numbering
Join	Joins two consecutive lines together
Key	Specifies the keyboard as the input source for CPT-TWX Model 33 and 35 terminals
LINesize	Sets session line length
LISt	Displays workspace contents
LOAD	Places a file in a workspace
local commands	In View Mode, work on line where entered:
nnnn/	Sets current line pointer
nnnnA	Adds lines
nnnnD	Deletes lines
nnnnR	Repeats lines
LOCate	Repositions current line pointer
MERge	Combines a file with workspace contents
MESsage	Sets conventions for message reception and message header printing
MOve	Repositions lines of data
NAME	Specifies an identifying name for a workspace
NEwline	Assigns logical new line character
NUmber	Specifies a sequence numbering convention for a file or a workspace
Off	Ends terminal session
PAssword	Specifies new logon password
PFkey	Specifies interpretation of 3270 terminal program function keys
PRotect	Specifies or removes file access restrictions
PUNch	Specifies paper tape output for CPT-TWX Model 33 and 35 terminals
PURge	Removes a file from a library
Query	Displays information about VSPC usage, libraries, or files
RELEase	Makes noncontrolled project library file available for transfer of ownership
RENumber	Changes workspace line numbers
ROute	Specifies an alternative destination for batch job output
RUn	Compiles and/or executes a source or object program, or a command list

Figure 14 (Part 1 of 2). VSPC Command Functions

Command	Action
SAve	Places workspace contents into a library
SCRatch	Erases job output
SEnd	Transmits messages from the terminal
SHare	Makes a file available to other users
SPlit	Separates one line into two consecutive lines
STatus	Queries system about status of a previously submitted batch job
STOre	Compiles a source program and places the object program in a library
SUBmit	Submits a job stream for batch processing
TABset	Sets session and logical tabs
TAPe	Specifies paper tape as input for CPT-TWX Model 33 and 35 terminals
TExt	Sets actions taken by CHANGE, FIND, and LOCATE commands
TRanslate	Sets upper/lowercase usage or usage of a translate table
View	Puts a display terminal in direct input/edit mode
VSPC	VTAM command to begin a terminal session

Note: Capital letters indicate the minimum abbreviation for the commands.

Figure 14 (Part 2 of 2). VSPC Command Functions

Command Syntax

The syntax of VSPC commands follows simple rules, outlined in the following paragraphs.

Each command name is a keyword that begins a line and must be the first nonseparator characters on that line.

The command name is optionally followed by one or more operands.

A separator must be specified between the command name and/or the operands. Valid separators are the blank, the comma, the tab (if tab settings are in effect), and the comment (`/* comment */`).

Operands can be keyword operands, which specify a predefined action that VSPC is to perform, or variable operands, which modify the command execution through user-supplied information.

If command operands allow a choice of two mutually exclusive keywords, more than one of the keywords may be specified. In this case, the last keyword specified is the one VSPC uses. For example:

```
protect * noread nowrite read
```

specifies that the READ option is to override the NOREAD option. (Note that if more than two options for a keyword are possible, then only one option may be specified, or the user gets an error message.)

Abbreviations of Keywords

Each VSPC keyword can be abbreviated to its significant characters by truncation from the right, as follows:

- The characters in a command must be unique among all commands. For example, the RENUMBER command can be abbreviated as RE. Other valid abbreviations are: REN, RENU, RENUM, RENUMB, and RENUMBE.
- The characters in a keyword operand must be unique among the keyword operands within the command.

Note: User-defined *processor-names* and *table-names* are considered to be VSPC keywords. Just like other VSPC keywords, they can be truncated from the right to their significant characters, with the following exception. If two such names exist so that one is an initial substring of the other (for example, BALL and BALLOT), then:

- Entry of anything longer than the shorter name is a reference to the longer name (in the example, BALLO is a valid abbreviation for BALLOT).
- Entry of the shorter name in full is a reference to the shorter name (in the example, BALL refers to BALL and not to BALLOT).
- Entry of an abbreviation for the shorter name is ambiguous and is rejected (in the example, B, BA, and BAL would be ambiguous abbreviations for either BALL or BALLOT).

An easy-to-remember abbreviation rule is that—except for user-defined table-names and processor-names—each VSPC keyword can always be abbreviated to its first three characters. For example, you can be sure that DEL is a valid abbreviation for the DELETE command.

Command Continuation

When the command *entry* ends with a minus sign (–), the next succeeding line is treated as a command continuation; logically, it immediately follows the last character preceding the minus sign. (Note that separators are considered to be characters.) For example, the following terminal entry:

```
QUERY–  
  FILE( FILEA )
```

is accepted as if entered as:

```
QUERY      FILE( FILEA )
```

When the command *entry* ends with a plus sign (+), the next succeeding line is treated as a field continuation; logically, the first nonseparator character of the succeeding line immediately follows the last character preceding the plus sign. For example, the following terminal entry:

```
QUERY FILE+  
  ( FILEA )
```

is accepted as if entered as:

```
QUERY FILE( FILEA )
```

| Commands entered in View Mode cannot be continued.

Comments

Comments in VSPC are valid. A comment must begin with the characters `/*` and end with the characters `*/`. For example:

```
/* this is a comment */
```

Comments do not affect the execution of the command or command list in which they are included.

Format Notation

Formats in this chapter are printed as shown in Figure 15. The meaning of the combinations of words, spaces, and special characters is explained in the following paragraphs.

COMMAND-NAME	<i>operand-1</i>	Explanatory information about operand-1.
	<i>operand-2</i> <i>operand-2</i> <i>operand-2</i>	Explanatory information about operand-2.
	<i>operand-3</i> <i>operand-3</i>	Explanatory information about operand-3.

Figure 15. Command Format Presentation

COMMAND-NAME

The command-name, in the lefthand column, is a VSPC keyword; it is shown in CAPITAL LETTERS.

The command-name must be spelled exactly as shown, except when it is abbreviated (see "Abbreviations of Keywords").

Operands

The operands valid for a command-name are listed in the middle column, using the following conventions:

- Operands enclosed between two horizontal lines represent available options for one operand. If all options may be omitted, the action taken is explained in the third column.
- Operands separated by a horizontal line represent separate operands. When specified, each must be separated from the next, and from the command-name, by a space, comma, or tab.
- Operands printed in CAPITAL LETTERS represent keyword operands, which identify predefined actions to be taken by VSPC. These operands, when specified, must be spelled exactly as shown, except when they are abbreviated (see "Abbreviations of Keywords"). Capitalized operands may have words in *italicized lowercase letters* following them; such words represent user-supplied information that modifies the keyword operand.

- Operands printed in *italicized lowercase letters* represent information supplied by the user. If the operand ends with a hyphen and a numeral, the basic meaning of the operand is unchanged. The suffix is added only to clarify text references.
- If the order in which operands are written is significant, an entry in the righthand column defines the required order. If no order is given, the operands may be written in any order.
- Options *within* operands must be written in the order shown.
- Special characters—some operands contain one or more of the following characters as part of the optional or required information:

Special Character	Meaning
()	encloses a data subfield
=	equates the logon request with a specific user number
'	encloses a character string
*	specifies either the workspace or the terminal
:	denotes a range of numbers
/	requests a password prompt

When such portions of the operands are entered from the terminal, these characters — except for the parentheses — must always be entered exactly as shown. For example:

Format	Valid Entry	Invalid Entry
<i>begin-line :end-line</i>	100:200	100 200
' <i>string</i> '	'host'	host

Note: Parentheses can be omitted from commands, except when their omission would result in an ambiguous interpretation of the command (as, for example, in the PASSWORD operand of the PROTECT command). In such cases, omission of the parentheses results in an error message. (However, in this manual, all examples show the parentheses for clarity.)

- Text references agree with the format notation. That is, a reference to a specific CAPITALIZED operand is CAPITALIZED; a reference to a specific *italicized* operand is *italicized*. For example: a reference to *line* is a reference to a specific *line* operand; a reference to “line”, however, is a reference to lines in general.

Explanatory Information

Explanatory information about each operand is given in the righthand column: if an operand is required or optional, if its position in relation to other operands is significant, if it is optional what action is taken when it is omitted, and so forth.

ACQUIRE Command

The ACQUIRE Command transfers ownership of a file in a noncontrolled project library.

ACQUIRE	<i>filename</i>	Required operand
---------	-----------------	------------------

filename

identifies the file to be transferred. The *filename* must conform to the rules for VSPC filenames. (See the "Filenames" and "Passwords" entries.)

General Execution Considerations

When the ACQUIRE command is executed, VSPC checks that the file resides in the user's noncontrolled project library, that the prospective owner has enough available space to store the file (the space occupied by the file is deducted from the current file space charged to the old owner and added to that of the new owner), and that the current owner has issued a RELEASE command.

If all these conditions are met, the user issuing the ACQUIRE command becomes the owner of the file and the released status is removed. If they are not met, the command is rejected, and the file transfer does not take place.

The library manager of the project library cannot use the ACQUIRE command to become the owner of files owned by other project users.

Note: The current owner of a file can use the ACQUIRE command to retract the result of a previous RELEASE command.

AID Commands

The AID commands provide online explanations of VSPC messages, online prompting in the formation of valid VSPC commands, and an online explanation of VSPC AID itself.

?	AID COMMANDS <i>command-name</i> <i>message-header</i> <i>n</i>	Optional—specify or omit. If omitted, command is equivalent to ?1.
---	---	--

AID

specifies an explanation of VSPC AID.

To this command, VSPC will reply with a message listing the following topics of information:

- Listing commands
- Building commands
- Explaining commands
- All of the above

For more information, the user should reply by entering either the number of the desired information, or no character. Entering no character terminates the explanation, and returns the user to normal VSPC

operation. If the user enters the number corresponding to all of the above, VSPC AID will reply with a message telling the user how to:

- Get a list of AID-supported commands
- Get prompting requests from AID (that is, AID requests that ask the user to enter information)
- Stop an AID prompting sequence
- Get an AID explanation of VSPC messages

COMMANDS

specifies a categorized list of valid VSPC commands.

To this command, VSPC AID will reply with the numbered list of VSPC command groups described in the explanation above.

command-name

specifies prompting in the formation of the VSPC command whose name is given.

For example, to the command “?LOAD”, VSPC AID will reply with a series of questions whose answers, supplied by the user, will specify the required and optional operands needed for a valid LOAD command.

message-header

specifies an explanation of the VSPC message whose header is given.

For example, to the command “?ASU220”, VSPC AID will reply with a further explanation of the standard message:

```
ASU220 UNPAIRED QUOTE
```

The explanation is the same as that given in the message explanations listing (see Appendix B).

n

is an integer from 1 through 8 specifying an explanation of the *n*th previous message sent to the user. “?” and “?1” both specify an explanation of the immediately preceding message.

General Execution Considerations

Inclusion of AID commands is optional at system installation time.

Not all messages will be explained by AID. Messages for which no further explanations are available include error messages sent by AID, and messages sent by AID to explain other messages.

AID gives the user four chances to reply fully to a prompt. Non-standard replies evoke an explanatory message and a repetition of the prompt; they include:

- replies containing syntax errors in a variable field (such as a six-digit number intended to be a line number),
- replies that do not match one of a set of options specified by the prompt, and

- the reply “?”.

After the fourth reply, the user returns to normal VSPC operation.

Under some circumstances the user can avoid completing an entire AID message or an entire sequence of prompts. If the user signals attention to cancel output during the operation of AID, the following actions are taken:

- If output was a message explanation or a list of commands in a command group, output ceases and normal VSPC operation resumes.
- If output was a prompt, output skips to the last line of the prompt (which asks a question), and AID waits for the reply.
- If output was the last line of a prompt, the first cancellation will cause the line to be displayed again. After a second cancellation, normal VSPC operation will resume.
- If VSPC was waiting for a reply to a prompt while forming a command, and if all remaining operands in the command are optional, the command as specified so far is displayed along with the final prompt that gives the user the choice of executing or canceling. If there are required operands yet to be specified, the user is given a message to that effect along with a prompt that allows the choice of completing the command or canceling it.

If an AID command is given during the processing of a command list, the requested VSPC AID function is performed. If the user forms a command and chooses to execute it, the command is executed and the next command is taken from the command list, as it is if the user terminates execution of AID in any other standard fashion. But if the user forms and chooses to execute a command that cannot be executed (for example, one referring to a non-existent file), VSPC will reject the command and also terminate processing of the command list.

ALLOCATE Command

The ALLOCATE command allows a VSPC processor user to assign a file to a processor unit-number. Processors for which the ALLOCATE command can be specified are the IBM program product VSPC FORTRAN and similar user-written compilers.

ALLOCATE	<i>filename</i> *	Optional first operand—choose one or omit. (Required if other operands are specified.)
	<i>unit-number</i>	Optional second operand. (Required if other operands are specified.)
	MOD	Optional—specify or omit. If omitted and first and second operands are specified, existing records overwritten.
	RUN SESSION	Optional—choose one or omit. If omitted and first and second operands are specified, RUN is assumed.
	If all operands are omitted, current values are displayed.	

filename or *

specifies the file or terminal assigned to the *unit-number*.

filename

specifies that the designated VSPC file is assigned to the processor *unit-number*; *filename* must conform to the rules for VSPC filenames (see the “Filenames” and “Passwords” entries).

*

specifies that the user’s terminal is assigned to the processor *unit-number*.

unit-number

specifies the processor unit-number. It must be an integer with a value from 1 through 99, inclusive. If a given *unit-number* has already been allocated, the new allocation replaces the former one.

Note: The *unit-number* is equivalent to the data set reference number used in FORTRAN input/output statements.

MOD

when specified, tells VSPC that current output data is to be added at the end of an already existing file.

MOD has meaning only for output files. If MOD is specified for a file being read, the option is ignored.

When MOD is omitted, the existing file is positioned for output at its beginning; that is, existing records are overwritten.

RUN or SESSION

specifies the length of time the allocation will stay in effect.

RUN

specifies that the *unit-number* allocation is to remain in effect until after the next RUN command is executed.

SESSION

specifies that the *unit-number* allocation is to remain in effect until the end of the current terminal session.

When both RUN and SESSION are omitted, RUN is assumed.

General Execution Considerations

When the ALLOCATE command is executed, the specified *unit-number* allocation goes into effect immediately.

A check on the validity of the file specification is made when the file is opened at execution time.

Note: A processor program can also issue *unit-number* allocations during program execution. If such a processor gives the user the facility to allocate *unit-numbers* (VSPC FORTRAN does), the user can use that facility instead of the ALLOCATE command, or to replace any existing ALLOCATE command *unit-number* allocations.

BACKSPACE Command

The BACKSPACE command specifies a character which, when entered into the workspace, is translated into the nonprintable backspace character.

BACKSPACE	'x' ' '	Optional—choose one or omit. If omitted, current value is displayed.
------------------	------------	---

'x'

specifies a character to be translated into the nonprintable backspace character.

' '

(two adjacent single quotation marks) specifies that no character is to be translated into the backspace character. (If a substitute character is currently specified, the specification is removed.)

General Execution Considerations

When the BACKSPACE command is issued, one of the two following actions takes place:

- If a substitution character is specified, the specified character is translated into the nonprintable backspace character whenever it is entered at the terminal and placed in the workspace.
- When two adjacent quotation marks are specified, no character translation takes place; any previous character translation is discontinued.

Note: VSPC does not treat the substitution character as a backspace for character correction.

CANCEL Job Entry Command

The CANCEL command requests that the system cancel an already submitted batch processing job.

CANCEL	<i>jobname</i>	Required operand.
	PURGE	In OS/VS, optional second operand; if omitted, output not erased.

jobname

identifies the job to be canceled. The *jobname* must conform to the rule for VSPC jobnames (see the "Jobnames" entry). Only one *jobname* may be specified.

The entire *jobname* may be specified; alternatively, the *jobname* may be specified omitting the 6-character job entry code.

PURGE

specifies, in OS/VS, that any job output already produced is to be erased.

General Execution Considerations

When the CANCEL command is executed, the VSPC job entry facility passes the request on to the batch processing system. If job processing has not already begun, the system cancels the job.

In OS/VS, the PURGE operand may be specified. When PURGE is specified:

- In OS/VS1, any existing job output is erased when the CANCEL command is executed, but system messages are not erased.
- In OS/VS2, both existing job output and system messages are erased when the CANCEL command is executed.

If PURGE is not specified and job execution is in progress, the job is canceled, but output already created is not erased.

If the job cannot be canceled, VSPC returns one of the following messages:

```
JOB NOT FOUND ON INPUT QUEUE  
JOB NOT CANCELLED, OUTPUT ON QUEUE
```

The CANCEL command is not applicable unless a previous SUBMIT command that includes this job has been executed (see the "SUBMIT Command" entry).

CHANGE Command

The CHANGE command causes a specific character sequence in the workspace to be edited.

CHANGE	<i>line</i> <i>line :end-line</i> * ALL	Required first operand—choose one.
	' <i>string-1</i> ' ' <i>string-1</i> ' ' <i>string-2</i> '	Required operand—choose one. (Optional for display terminals.)
	TEXT NOTEXT	Optional—choose one or omit. If omitted, NOTEXT is assumed, unless TEXT ON was specified. If specified, must follow string operand.

line and *end-line*

specify integers that correspond to line numbers in the workspace.

line

when specified alone, represents a single line to be changed; the line number specified must actually exist in the workspace.

line :end-line

when specified, represent a range of lines to be searched. If the integers specified are not actual line numbers, all existing lines in the range specified are searched. The value of *end-line* must be equal to or greater than that of *line*.

*

specifies the current line. It can be specified in place of *line*, *end-line*, or both.

ALL

specifies that all lines of editable data in the workspace are searched.

string-1 and *string-2*

specify the characters to be used; each may consist of 0 through 255 characters enclosed in single quotation marks; *string-1* and *string-2* need not contain the same number of characters. If the strings contain only alphabetic and numeric characters, then the quotation marks may be omitted. However, if quotation marks are omitted, lowercase characters in the string will be automatically translated to uppercase for command entry. To get a string containing lowercase characters, specify TRANSLATE OFF and enclose the string in quotation marks.

All characters are valid. However, if a single quotation mark (') is contained within either string, it must be specified as two adjacent single quotation marks (' '). In the workspace, the adjacent quotation marks are treated as one single quotation mark.

string-1

specifies the characters to be changed.

string-2

specifies the replacement characters.

Any number of separators may be placed between *string-1* and *string-2*.

For display terminals, if all the operands are omitted after a single line number (or *), the line is copied in the command area. This allows editing before pressing ENTER. This function is not supported from a CLIST.

TEXT or NOTEXT

specifies the display convention to be used.

TEXT

specifies that changed lines are to be displayed at the terminal.

NOTEXT

specifies that only the line numbers of changed lines are to be displayed.

When both TEXT and NOTEXT are omitted, NOTEXT is in effect, unless a TEXT ON command has been previously executed (see TEXT Command entry).

General Execution Considerations

When the CHANGE command is executed, all occurrences of *string-1* within the specified range of lines are replaced by *string-2*. The following replacement rules apply:

- If *string-1* only is specified, then *string-1* is deleted. (*string-2*, when omitted, is considered to contain 0 characters.)
- If both *string-1* and *string-2* are specified, *string-1* is replaced by *string-2*.
- If *string-1* is entered as having 0 characters (that is, if *string-1* is entered as two adjacent single quotation marks (' '), and *string-2* is specified, then *string-2* is added at the end of the specified lines.
- If both strings are entered as having 0 characters, all lines or line numbers—depending on whether TEXT or NOTEXT is in effect—within the specified range are displayed.

When execution is successful and TEXT is in effect, all changed lines are displayed at the terminal.

When execution is successful and NOTEXT is in effect, only the line numbers of changed lines are displayed.

In either case, VSPC responds with the following message:

nnn LINES CHANGED

where *nnn* is the number of lines which have been changed.

CLEAR Command

The CLEAR command provides an empty workspace.

CLEAR		No operands allowed.
--------------	--	----------------------

When the CLEAR command is executed, VSPC erases all lines from the workspace, and also removes the following associated items, if present:

- the *filename* (including the password)
- any PROTECT specification
- any SHARE specification
- any NUMBER specification differing from the default

The workspace attribute (whether the default attribute or one specified through an ENTER command) is retained.

COPY Command

The COPY command duplicates an existing line or range of lines in another location in the workspace.

COPY	<i>line</i> <i>line:end-line</i> *	Required first operand—choose one
	<i>destination-line</i> *	Required second operand
	<i>increment</i>	Optional third operand—specify or omit. If omitted, 10 is assumed.

line and *end-line*

must be integers within the range 0 through 99999.

line

specifies the present line number of a single line to be moved. The line number specified must actually exist in the workspace.

line:end-line

specifies the present position of a range of lines to be duplicated; *line* specifies the first line in the range, and *end-line* specifies the last. The line numbers specified need not actually exist in the workspace; if they do not, then existing lines within the range specified are duplicated. The *end-line* must be equal or greater in value than *line*.

destination-line

specifies the line-number after which the lines are to be duplicated. If this line-number exists, the lines are copied following this line. If this line-number does not exist, this number becomes the line-number of the first copied line.

*

specifies the current line. It can be specified in place of *line*, *end-line*, or *destination-line*.

increment

specifies the increment by which the duplicated lines are to be numbered. It must be an integer in the range 1 through 99999. When *increment* is omitted, an increment of 10 is assumed.

General Execution Considerations

When the COPY command is issued, the specified line or lines are duplicated following the specified destination.

If the newly inserted lines cause line-number overlap, the following lines are renumbered, using the same *increment* as the inserted lines, as far as necessary to maintain line-number sequence.

If the destination line is within the range of lines to be duplicated, the command is rejected.

DELETE Command

The DELETE command erases one or more lines from the workspace.

DELETE	<i>line</i> <i>line:end-line</i> *	Required operand—choose one.
---------------	--	------------------------------

line and *end-line*

must both be integers that specify line numbers within the workspace.

line

specifies that this one line is to be erased. The line specified must actually exist in the workspace.

line:end-line

specifies that the range of lines between *line* and *end-line* is to be erased. In this case, neither of the specified lines need actually exist in the workspace.

When *end-line* is specified, it must identify a line number equal to or greater than *line*.

*

denotes the current line number. * may be substituted for *line*, *end-line*, or both.

When execution is successful, the following message is printed:

nnn LINES DELETED

where *nnn* is the number of lines which have been deleted.

ENTER Command

The ENTER command explicitly specifies the attribute of the workspace.

ENTER	<i>processor-name</i> CLIST DATA	Optional—choose one or omit. If omitted, attribute of current workspace is displayed.
	' <i>string</i> '	Optional

processor-name

specifies a workspace attribute associated with one of the IBM program products that run under VSPC, or a user-written compiler or interpreter so designed. For the IBM program products, valid *processor-names* are shown in Figure 16.

Program Product	Processor-name(s)	Comments
VS BASIC	BASIC	Compiler—short precision
	LBASIC	Compiler—long precision
VSPC FORTRAN	FORTRAN	Compiler—free format
	XFORTRAN	Compiler—fixed format
	FLINK*	Link processor
VS APL	APL	Interpreter

*Available in OS/VS only.

Figure 16. IBM Program Products and Valid Processor-names

When *processor-name* is specified, the named processor must be installed with VSPC. If the processor is:

- an *interpreter*, an empty unnamed workspace is provided, and the user is placed under the interpreter's control. (This means that interpreter commands (rather than VSPC commands) control execution of the interpreter.)
- a *compiler*, the user remains under control of VSPC, with the contents of the workspace unchanged. (This means that a source program can be entered as line-numbered data, and that VSPC commands are used to compile, save, and run that program.)

CLIST

specifies a workspace that can contain VSPC command lists (see the "Processing Command Lists" section).

DATA

specifies a workspace attribute that is not associated with any specific processing capability. The workspace can be used for any form of editable data.

When the ENTER command is executed, the new attribute replaces any previous attribute for this workspace.

'string'

specifies a character string to be passed to the interpreter if processor-name specifies an interpreter. If processor-name specifies a compiler, the string is ignored.

EXTRACT Command

The EXTRACT command erases all but a specified line or range of lines in the workspace.

EXTRACT	<i>line</i> <i>line : end-line</i> *	Required operand—choose one.
----------------	--	------------------------------

line and *end-line*

must be integers within the range 0 through 99999.

line

specifies the line number of a single line to be retained. The line number specified must actually exist in the workspace.

line: end-line

specifies a range of lines to be retained; *line* specifies the first line in the range, and *end-line* specifies the last. The line numbers specified need not actually exist in the workspace; if they do not, then existing lines within the range specified are retained. The *end-line* must be equal or greater in value than *line*.

*

specifies the current line. It may be specified in place of *line*, *end-line*, or both.

General Execution Considerations

When the EXTRACT command is executed, all of the existing workspace, except for the specified line or range of lines, is erased. The line-numbers of the retained lines are unchanged.

The current line pointer is set as follows:

- If the current line is within the range of the retained lines, the current line pointer is unchanged.
- If the current line precedes the retained lines, the current line pointer is set to the first retained line.
- If the current line follows the retained lines, the current line pointer is set to the last retained line.

The name of the workspace is erased; the workspace attribute is unchanged.

FEATURE Command

The **FEATURE** command tells **VSPC** whether or not to use an **APL/Text** feature on your **IBM 3270 Information Display System** terminal.

FEATURE	APL NOAPL	Optional operand - choose one or omit. If omitted, current value is displayed.
----------------	----------------------------	--

APL

specifies that **VSPC** is to assume that your terminal has an **APL/Text** feature and is to use it for output to your terminal.

NOAPL

specifies that **VSPC** is not to use an **APL/Text** feature for output to your terminal.

General Execution Considerations

This command is valid only when your terminal is an **IBM 3270 Information Display System** terminal. For full output of certain national character sets (such as **Katakana** or **Canadian/French**), it is necessary to specify **FEATURE NOAPL**, even if your terminal actually has an **APL/Text** feature.

FILE Command

The FILE command creates files or modifies the characteristics of existing files.

FILE	<i>filename</i>	Required first operand.
	<i>n</i>	Optional second operand. If omitted for new files, editable size limit assumed. If omitted for existing files, present size limit assumed.
	SEQUENTIAL DIRECT	Optional—choose one or omit. If omitted for new files, SEQUENTIAL is assumed. If omitted for existing files, the present organization is assumed.
	UNDEFINED	Optional—specify or omit. If omitted, file is editable.
	CONTENT (<i>attribute</i>)	Optional—specify or omit. If omitted for new files, DATA attribute assumed. If omitted for existing files, content attribute is unchanged.

filename

identifies the file to be created or modified; it must conform to the rules for VSPC filenames (see the “Filenames” and “Passwords” entries).

The *filename* must not specify the special file **CONTINUE**.

n

specifies, in thousands of bytes, the maximum size permitted for this file; when specified it must be an integer in the range 0 through 65,535.

SEQUENTIAL or **DIRECT**

when specified, identifies the file organization for this file.

SEQUENTIAL

identifies this file as sequentially organized.

DIRECT

identifies this file as a direct file.

When this operand is omitted for new files, **SEQUENTIAL** is assumed. When it is omitted for existing files, the present organization is assumed.

UNDEFINED

identifies this file as a foreground processor data file that cannot be edited by VSPC. The file organization can be sequential or direct; if neither the **SEQUENTIAL** or **DIRECT** operand is specified, **SEQUENTIAL** is assumed.

(See the "File Organization" entry for a complete description of each type of file organization.)

CONTENT (*attribute*)

specifies the content attribute of the file. The *attribute* may be any one of the *processor-names* shown in Figure 15, or it may be CLIST or DATA. If CONTENT is omitted for new files, the DATA attribute is assumed; if it is omitted for existing files, the content attribute is unchanged.

General Execution Considerations

When the FILE command is executed, the file identified by *filename* is either created or modified.

File Creation

If the *filename* does not already exist, a directory entry with the specified or implied attribute is created for it, but no space is allocated to it.

In this case, *n* specifies the maximum size for this file. If *n* is omitted, this user's size limit for editable files is assumed.

File Directory Entry Modification

If the file already exists, the following warning message is sent:

WARNING - FILE ALREADY EXISTS

and the FILE command is executed. (At least one operand in addition to *filename* must be specified.)

In this case, *n* specifies the new maximum size allowed for this file. If the contents of the file are larger than *n*, the following warning message, instead of the previous one, is sent:

WARNING - FILE LIMIT LESS THAN EXISTING FILE SIZE

The limit will then be set to the value of *n*. The size of present file can then no longer be increased, and, if it is ever rewritten, *n* will become the new maximum file size. (Note, therefore, that zero (0) is a legal maximum file size specification.)

If *n* is omitted, the present maximum size is unchanged.

The SEQUENTIAL, DIRECT, and/or UNDEFINED operand respecifies the file organization. If the operand is omitted, the present file organization is unchanged.

If DIRECT is specified, and the existing file does not meet the requirements for a direct file (see the "File Organization" entry), the command is rejected.

A nonempty sequential or direct file cannot be changed to an undefined sequential or direct file, or vice-versa.

An object file cannot be changed to any other type.

File Organization

There can be five types of files in a VSPC library: sequential, direct, object program, undefined sequential, and undefined direct. In addition, some processors may allow VSAM files outside the VSPC library to be accessed.

Each type of file organization, and, where applicable, the type of records associated with it, is discussed in the following paragraphs.

Sequential Files

A VSPC sequential file is one in which the records are positioned in line number order. Foreground processors can then retrieve the records in that same order. Records in the file can be of varying lengths.

A VSPC sequential file must consist of editable data; that is, the file can have one of the following attributes: DATA, CLIST, or *processor-name* (when the file consists of compiler source statements). Each record is identified by a line number. When the file is loaded into the workspace, records in the file can be edited (that is, modified, rearranged, added, or deleted). VSPC commands can be used to edit the records in any order.

Direct Files

A VSPC direct file is one in which references to specific records can be made directly through logical record number (which, for direct files, is the same as the line number). In a VSPC direct file, records are positioned in line number order. Records can then be retrieved either sequentially in this same order, or specific records can be directly retrieved, using as identification the logical record number.

A VSPC direct file must have all the following specifications:

- The file attribute must be DATA.
- Line numbers must begin at 1 and increment by 1.
- All records must be the same length.
- The total number of logical records in the file must be no greater than:

$$1015 * (4063 / (size + 5))$$

where *size* is the length of the data portion of each logical record. The *size* may be any integer from 0 through 4058.

VSPC direct files are used by some VSPC processors as data files from which records can be either sequentially or directly retrieved. VSPC commands can be used to edit the records of a direct file in any order.

Object Program Files

An object program file is created when the user saves a compiled object program or an interpreter workspace.

Such a file consists of machine language object program or interpreter instructions. Object program files are made available to the VSPC user through the RUN command (see the "RUN Command" entry) or through interpreter facilities. They cannot be edited.

Undefined Files

An undefined sequential or direct file is a special purpose data file used by a foreground processor program (as, for example, a BASIC or FORTRAN object program); the data format in such files is defined by the foreground processor and is presumed to be one that VSPC cannot edit. Thus, undefined files are processed only by foreground processors; VSPC cannot edit them. However, the VSPC library commands and job entry commands can be used with them.

External VSAM Files

External VSAM files (that is, files with VSAM organization stored outside the VSPC library) can be made available directly to foreground processors through one specific library number. All such files are accessed through that library number.

Entry Sequenced Data Sets (ESDS) can be processed through their order of entry or through alternate indexes.

Key Sequenced Data Sets (KSDS) can be processed through primary or alternate indexes.

Relative Record Data Sets (RSDS) can be processed through their relative record numbers.

The specific formats and processing options allowed depend upon the processor in use. VSPC itself does not process VSAM files.

Filenames

VSPC *filenames* are used to refer to files in VSPC libraries. A fully qualified *filename* has the following form:

libnum name/password

libnum

is a numeric identifier one through seven digits long that identifies one specific VSPC library. It is the user number of the library manager.

For a public library file, *libnum* is required. *libnum* must be the user number of the public library manager.

For a project library file, *libnum* is required. *libnum* may be the user number of the project library manager; or it can be specified as zero (0).

For a private library file, *libnum* is the user number of the private library manager.

When the reference is to the user's own library—public, project, or private—*libnum* can be omitted.

name

identifies one specific file within a specific library. The *name* must contain one through eight alphanumeric characters (A through Z, 0 through 9, \$, #, @); the first character must be alphabetic. (Note that certain foreground processors have different rules for *name*; see the appropriate terminal user's guide for details.)

/password

is specified with *filename* only if the related file is to be, or has been, password protected. The form of a *password* is given in the "Passwords" entry.

No separator is required between *libnum* and *name*; however, one or more may be specified. No separator is required between *name* and */password*; however, one or more may be specified.

Both the *libnum* field and the *password* remain the same until explicitly changed through execution of a LOAD, NAME, SAVE, or PROTECT command. For example, if user 12345 executes the following sequence of commands:

```
LOAD 22222ADDPROG/SQUELCH
SAVE
```

the SAVE command will save the file back into library 22222 as ADDPROG with the password SQUELCH. However, if user 12345 executes the following sequence of commands:

```
LOAD 22222ADDPROG/SQUELCH
SAVE 12345NEWPROG/SUBROSA
```

when the SAVE command is executed, the old program is stored in his own library as NEWPROG with the new password SUBROSA, and the workspace name is correspondingly changed. (Note that the same program still exists in library 22222 as ADDPROG with the password SQUELCH.)

Omission of *libnum* and/or */password* can be an explicit change. For example, if user 12345 executes the following sequence of commands:

```
LOAD 22222ADDPROG/SQUELCH
SAVE NEWPROG
```

The SAVE command places the file in his own library without a password, and the workspace name is correspondingly changed to 12345NEWPROG without a password.

Special File CONTINUE

Any VSPC library can contain a special file with the *filename* CONTINUE. This file is used by VSPC to ensure that current work in the user's workspace is automatically saved, under certain circumstances, at the end of a terminal session.

When the workspace contains editable data or interpreter statements, the CONTINUE file is created when either:

- The user issues an OFF CONTINUE command (or an interpreter equivalent) at the end of the session.
- The session is ended by conditions beyond the user's control (what is called a "forced ending").

In either case, the saved work can be made available during the next session with the following command (or its interpreter equivalent):

```
LOAD CONTINUE
```

CONTINUE usage conforms to the following rules:

- The user can retrieve a CONTINUE file only from his own library.
- Unless the user specifies OFF CONTINUE, a CONTINUE file is automatically removed from his library when the next OFF command is executed.
- A CONTINUE file can be overlaid by a CONTINUE file of another type.
- The SAVE command can specify CONTINUE. Note, however, that this could accidentally destroy data already present in the CONTINUE file—data automatically saved at the end of a previous terminal session. Note, also, that any file saved with the name CONTINUE will normally be purged at the end of the present session.
- The FILE, NUMBER, PROTECT, SHARE, STORE, and SUBMIT commands may not specify CONTINUE.
- The CONTINUE file cannot be used for processing as a data file in any mode.

FIND Command

The FIND command locates occurrences of a specific character sequence within the workspace.

FIND	<i>line</i> <i>line:end-line</i> *	Optional first operand—choose one or omit. If omitted, entire workspace is searched.
	'string '	Required operand.
	TEXT NOTEXT	Optional—choose one or omit. If omitted, NOTEXT is assumed. If specified, must follow 'string' operand.

line and *end-line*

specify integers that fit within the existing range of line numbers in the workspace.

line

when specified alone, represents a single line to be searched; the line number specified must actually exist in the workspace.

line:end-line

when specified, represents a range of lines to be searched. If the integers specified are not existing line numbers, all actual lines in the range specified are searched. The value of *end-line* must not be less than that of *line*.

When both *line* and *end-line* are omitted, all lines of editable data in the workspace are searched.

*

denotes the current line. * may be specified instead of *line*, *end-line*, or both.

'string '

represents the characters searched for; it must consist of 0 through 255 characters enclosed in single quotation marks. If the string contains only alphabetic and numeric characters, then the quotation marks may be omitted. However, if quotation marks are omitted, lowercase characters in the string will be automatically translated to uppercase for command entry. To get a string containing lowercase characters, specify TRANSLATE OFF and enclose the string in quotation marks.

All characters are valid. However, if a single quotation mark (') is contained within the string, it must be specified as two adjacent single quotation marks (' '). In the workspace, the adjacent quotation marks are treated as one single quotation mark.

TEXT and NOTEXT

specify the printing convention to be used.

TEXT

specifies that each line in which *string* occurs is to be displayed at the terminal.

NOTEXT

specifies that a table of the line numbers in which *string* occurs is to be displayed at the terminal.

When both TEXT and NOTEXT are omitted, NOTEXT is assumed, unless a TEXT ON command has been previously executed (see TEXT Command entry).

FREE Command

The FREE command allows a foreground processor user to remove a previously-specified allocation of a processor unit-number. (See the "ALLOCATE Command" entry.)

FREE	<i>unit-number</i>	Optional operand. If omitted, all allocations are removed.
-------------	--------------------	--

unit-number

specifies a previously-allocated unit-number. When specified, it must be an integer with a value from 1 through 99, inclusive.

When the FREE command is executed, previous allocations of processor *unit-numbers* are removed according to the following rules:

- If *unit-number* is specified, only the allocation of the specified *unit-number* is removed.
- If *unit-number* is omitted, all of this user's *unit-number* allocations are removed.

HARDCOPY Command

For IBM 3270 Information Display System users, the HARDCOPY command specifies a printer *node-name* to be used in making copies of the display screen or relinquishes the use of a printer.

HARDCOPY	<i>node-name</i> * OFF	Optional operand—choose one or omit. If omitted, current value is displayed.
-----------------	-------------------------------------	--

node-name

requests the use of a separate 3270 Display System printer. The *node-name* must be predefined by the system administrator.

*

indicates that a 3284 printer is directly attached to a 3275 Display Unit, and that this printer is to be used.

OFF

specifies that the printer currently in use is to be disconnected.

The HARDCOPY command is valid only when entered from a 3270 terminal.

For such terminals, the HARDCOPY command must be executed before printing operations can be requested.

Printing of the current page is requested by positioning the cursor at the HCPY field on the control line and pressing the ENTER key, by positioning the light pen at the HCPY field, or by use of a program function key predefined as HCPY. The current page is then printed.

INPUT Command

The INPUT command begins automatic line numbering for line entry.

INPUT	<i>begin-line</i> <i>begin-line increment</i> *	Optional first operand—choose one or omit. If omitted, for empty workspace line 10 assumed; for nonempty workspace next higher multiple of 10 than last existing line assumed; increment 10 assumed.
	PROMPT NOPROMPT	Optional—choose one or omit. If omitted, PROMPT is assumed.
	OVERLAY	Optional third operand—specify or omit. If omitted, existing lines not replaced.

begin-line and *increment*

specify the line numbering conventions to be used.

begin-line

specifies the first generated INPUT line number. When specified, it must be an integer from 0 through 99999, inclusive.

increment

specifies the increment by which succeeding line numbers are generated. When specified, it must be an integer from 1 through 99999, inclusive. When *increment* is omitted, an increment of 10 is assumed.

*

denotes the current line. * may be specified instead of *begin-line*, whether or not *increment* is specified.

PROMPT and NOPROMPT

specify the display convention to be used.

PROMPT

specifies that automatically generated line numbers will be displayed at the terminal.

NOPROMPT

specifies that automatically generated line numbers will not be displayed at the terminal.

When both PROMPT and NOPROMPT are omitted, PROMPT is assumed.

OVERLAY

specifies that lines entered in INPUT mode may replace existing lines in the workspace.

General Execution Considerations

When *begin-line* is specified and OVERLAY is not, and the workspace contains lines, VSPC checks for overlap between the beginning line number and existing workspace lines. If two line numbers are identical, VSPC renumbers the old lines as far as is necessary to retain line sequencing.

When *begin-line* and OVERLAY are both specified and the workspace contains lines, VSPC makes no check for overlap between the beginning line number and existing workspace line numbers. Thus, it is possible that current lines of data will be overwritten. For example, if the last existing line number is 220 and INPUT 190 is specified, then any original lines from 190 through 220 may be replaced by data entered during INPUT mode.

When *begin-line* is omitted, and the workspace does not contain any lines, the first generated INPUT line number is 10; the assumed increment is 10.

When *begin-line* is omitted, and the workspace contains line entries, the first generated INPUT line-number is the multiple of 10 next higher than the current line number; the assumed increment is 10.

When PROMPT is in effect, each generated line number followed by a space is displayed at the terminal and the terminal is ready for input. The generated line number cannot be modified by editing. When the ENTER key is pressed before any line entry data, automatic line numbering ends. The terminal is then ready for input, but no line number is generated.

When NOPROMPT is specified, the following message indicates the starting line number:

LINE *line-number*

where *line-number* is the first generated INPUT line number. The terminal is now ready for input. For succeeding INPUT line entries, line numbers followed by a blank are automatically generated, but are not displayed at the terminal. Automatic line numbering ends when the ENTER key is pressed (also attention on typewriter terminals) before any line entry data. The following message is printed:

INPUT ENDED AT LINE *line-number*

where *line-number* is the the last number assigned. However, if no lines have been entered, the response is READY. The next command may now be entered.

See the section on line entry command mode for additional information.

Jobnames

In job streams created by the SUBMIT command, *jobnames* are generated by VSPC for each separate job in the stream, and serve to identify the job in subsequent VSPC commands.

A VSPC *jobname* always has the form *xxxxxyy zzzz*, where:

xxxxxx

is the job entry code specified in this user's profile.

yy

is a two-character identification code generated by VSPC. For each user, the code is assigned in ascending order, as follows: 01, 02, 03, ... 99, 00, 01,

zzzz

is a four-digit job number assigned by JES2 or JES3. This number is present in *jobnames* only under OS/VS2 VSPC. This number may be omitted if it is unavailable, but its omission will cause job entry commands to take significantly longer to execute.

If there are user-supplied job statements in the job stream, VSPC replaces the user-supplied jobname with one generated by VSPC. Therefore, the user-supplied job statement must reserve enough space to allow the eight-character VSPC *jobname* to be inserted.

VSPC *jobnames* are generated when a SUBMIT command is executed.

When a job is successfully submitted, VSPC returns the following message:

JOB SUBMITTED AS *jobname*

In subsequent CANCEL, LOAD, ROUTE, SCRATCH, or STATUS commands, the user can specify the complete *jobname*, or can optionally omit the 6-character job entry code.

If a job statement is detected following a DD DATA statement, VSPC returns the following message:

JOB CARD IN DATA HAS BEEN ASSIGNED JOBNAME *jobname*

In DOS/VS only, VSPC assigns only one *jobname* to the job stream when the SUBMIT command is executed. It does not modify any other DOS/VS JCL or POWER/VS JECL job cards.

The VSPC *jobname* must be used when the CANCEL, LOAD, ROUTE, SCRATCH, and STATUS job entry commands are specified.

JOIN Command

The JOIN command combines two consecutive lines in the workspace into one line.

JOIN	<i>line</i> *	Required operand—choose one
------	------------------	-----------------------------

line

specifies the line-number of the first of the two lines to be combined.

*

specifies the current line. It may be specified in place of *line*.

General Execution Considerations

When the JOIN command is executed, the specified line and the next succeeding line in the workspace are combined into one line.

The resulting line has the line-number of the first combined line.

The line-number of the second line is erased; succeeding lines are not renumbered.

KEY Command

For CPT-TWX terminals, the KEY command specifies the keyboard as the input source.

KEY		No operands allowed.
-----	--	----------------------

At logon, the keyboard is the assumed input source. Whether or not terminal input is read from the keyboard depends on the manual setup of the CPT-TWX terminal. If the manual setup is wrong, input is incorrectly formatted on the console sheet.

When the KEY command is in effect, VSPC automatically causes a line feed after each input line is entered.

The KEY command is valid only from CPT-TWX terminals.

Note: The TAPE command specifies the paper tape reader as the input source, reversing the effect of the KEY command.

Libraries

In VSPC, when the VSPC administrator creates a user profile, an associated user library is also created. The library can be used to save programs and other data for future reference. Within each library, individual items are saved as *files* and in commands are identified by their *filenames*. Filenames are explained fully in the "Filenames" entry. In VSPC, all references to files are made through their *filenames*.

The user's *libnum* (which is the same as his user number) always serves to identify a specific user library within the VSPC library. The user whose *libnum* identifies the user library is defined as the *library manager* for that library.

In any library, the user who creates a specific file is defined as the *owner* of that file. Only the owner can modify a file. Only the owner or the library manager can remove a file.

The user profile may specify the associated user library as one of the following types: private, project, or public. Each is explained in the following paragraphs.

Private Libraries

A private library is meant to be used primarily by one user. Therefore, the *library manager* and the *owner* of all the files are the same user, who can create, modify, and remove files. If this user specifies, through the SHARE command, that a specific file can be accessed by other users, then the other users are allowed only to access that specific file. (See Figure 17 for further information on file access.)

Project Libraries

A project library is one that is accessible to a specified group of VSPC users (the group is defined through the user profile). Within the project library group, a reference to the library manager's *libnum* or a reference to a *libnum* of 0 (zero) is a reference to this project library. Within a VSPC system, there can be more than one project library; however, in addition to his own library, any VSPC user may have access to only one project library.

A project library can be defined so that:

- Only the *library manager* is allowed to create, modify, and remove files within it. In this case, other users in the project group are restricted to reading the files. This type of project library is defined as *controlled*.
- Any user in the project group can create files. In this case, only the *owner* can modify the file; other users (including the *library manager*) cannot. The ownership of a file can be transferred to another user in the project group through the RELEASE and ACQUIRE commands (see the RELEASE Command and ACQUIRE command descriptions). The *owner* and the *library manager* can remove a file from the project library; all other users in the project group may only read it. (See Figure 17 for information on file access.) This type of project library is defined as *noncontrolled*.

Access Command	Library Type		
	Private	Project	Public
no access control commands	no other user can access	all project members can read	all users can read
SHARE YES	all other users can read	all users can read (as well as project members)	all users can read (as they always can)
PROTECT NOREAD	no other user can access	nonowners in project can load and run but not read	nonowners can load or run but not read
PROTECT NOWRITE	no other user can access; owner can't modify	all project members can read; owner can't modify	all users can read; owner can't modify
PROTECT NOREAD PROTECT NOWRITE	no other user can access; owner can't modify	nonowners in project can load or run but not read; owner can't modify	nonowners can load or run but not read; owner can't modify
SHARE YES PROTECT NOREAD	nonowners can load or run but not read	nonowners can load or run but not read	nonowners can load or run but not read
SHARE YES PROTECT NOWRITE	all users can read; owner can't modify	all users can read; owner can't modify	all users can read; owner can't modify
SHARE YES PROTECT NOREAD PROTECT NOWRITE	nonowners can load or run but not read; owner can't modify	nonowners can load or run but not read; owner can't modify	nonowners can load or run but not read; owner can't modify

Figure 17. Library Types and Access Control Commands

Public Libraries

A public library is one that is accessible to all VSPC users.

A public library can be defined so that:

- Only the *library manager* can create, modify, and remove files within it. Other VSPC users can only read these files. This type of public library is defined as *controlled*.
- Any VSPC user can create files. In this case, only the *owner* can modify the file; other VSPC users (including the *library manager*) cannot. The *owner* and the *library manager* can remove the file from the public library; other VSPC users can only read it. (See Figure 16 for information on file access.)

(See also the “Filenames” and “File Organization” entries.)

Line Entry—Command Mode

Command Mode line entry adds editable data to the workspace. Command Mode line entry is the only form of line entry available to typewriter terminals. On display terminals, Command Mode line entry is entered on the screen command area in the following form:

nnn entry

nnn

represents a line number, one through five digits long. Leading zeros need not be supplied.

When *nnn* is less than five digits long and the first character of the *entry* is numeric, a space must be inserted between *nnn* and *entry*. Otherwise the space may be omitted.

Blank characters can precede the first character of the line number.

The user can enter line numbers or use the INPUT command to get line numbers automatically.

entry

can be any editable data. That is, it can be a command list entry (see the “Processing Command Lists” section), a line in a compiler source program, or, if the attribute is DATA, it can be simple editable data.

When originally entered from the terminal, the maximum line length, including the line number, is 1,000 characters, which allows ample space for backspacing and underscoring.

The entry can contain any graphic characters. Except when it is used for character deletion, the backspace character (or its substitute character) is a valid data character in the line entry.

The line entry is entered in the same way as a VSPC command.

General Execution Considerations

When Command Mode line entry is successful, the response is to accept further line or command entries.

When Command Mode line entry is unsuccessful, an error message is sent to the terminal.

Line Entry—VIEW Mode

For display terminals, View Mode line entry adds editable data to the workspace in the bulk input area (lines 1-20 on the display screen).

The VIEW command followed by the INPUT command makes View Mode line entry available. (See entries for the VIEW and INPUT commands.)

When INPUT PROMPT is specified, VSPC displays the current line at the top of the screen and generates line numbers in the first 5 positions of the remaining lines of the bulk input area. (The line numbers used are calculated from the *begin-line* and *increment* operands of the INPUT command. The remainder of

each line is blank; the user can then enter data in any character position beyond the sixth, by positioning the cursor at the desired line and typing in data. No line entered can be longer than 73 characters.

When INPUT NOPROMPT is specified, VSPC displays the current line at the top of the screen and the remaining lines of the bulk input area are blank; all character positions within that area are available for data entry. The user can specify the end of a line through a substitute NEWLINE character (see the NEWLINE Command entry). When VSPC receives the entry, the new line character is used to end the current line, to create a line number for the next line, and to begin the next line. The line numbers are calculated from the *begin-line* and *increment* operands of the INPUT command.

View Mode Processing Procedures

The order in which VSPC processes requests depends on the way in which they are entered:

1. If the light pen is used to select control-line fields, nothing else on the screen is processed.
2. If the PFKEY NOENTER command is used to set a character-string, nothing is changed in the workspace until the user executes another command or presses the ENTER key.

If the PFKEY ENTER command is used to set a character-string, it's as if the character-string were placed into the screen position of the cursor and the ENTER key pressed.

3. If HCPY is requested through a preset PFKEY or through cursor positioning followed by ENTER, changes on the screen are made in the workspace, then the screen is printed incorporating the changes.
4. If the PA1 key is used to signal attention, nothing else on the screen is processed.
5. If the PA2 key is used, output is canceled.
6. If CLEAR is used, any screen changes are canceled, and the screen is redisplayed in its original form. Followed by ENTER, changes on the screen are made in the workspace; then the screen is printed, incorporating the changes.
7. When the ENTER key is pressed, screen changes are placed in the workspace in the following order:
 - a. line changes and local commands, beginning at the top of the screen and continuing to the bottom see also the Local Commands entry
 - b. general commands entered in the command area
 - c. forward or backward paging requested by placing the cursor in the control line and pressing ENTER.
 - d. a PFKEY preset to FORWARD or BACKWARD.

General Execution Considerations

As the data is entered, the user can make corrections, additions, and deletions through the terminal editing keys (INS MODE, DEL, ERASE EOF, etc.) and through the VSPC local and general commands.

Each time the user presses the ENTER key, the contents of the bulk input area (screen lines 1-20) are transmitted to the workspace and the screen is

reformatted for additional input.

To end View Mode line entry, the user presses the ENTER key before making any changes on the current screen.

When originally entered from the terminal, the maximum line length, including the line number, is 1000 characters. This allows ample space for backspacing and underscoring.

The line entry can contain any graphic characters. Except when it is used for character deletion, the backspace character (or its substitute character) is a valid character in the line entry.

When View Mode line entry is successful, the response is to accept further line or command entries.

LINESIZE Command

The LINESIZE command specifies the session line length for the terminal.

LINESIZE	<i>n</i>	Optional first operand—specify or omit.
	<i>x</i>	Optional second CPT-TWX operand—specify or omit. If omitted and first operand is specified, idles specification unchanged.
	If both operands are omitted, current linesize is displayed.	

n specifies the number of character positions to be used for a display line at the terminal; it must be an integer with a value from 18 through 255 inclusive, except for terminals that allow paper tape input and output, for which *n* may have a value from 18 through 32767, inclusive.

x is valid only for CPT-TWX terminals. It must be specified as an integer between 0 and 100, inclusive. When this operand is specified, *x* number of extra idles are inserted into the terminal output after each carrier return. This can be used to ensure enough idle time for the actual physical carrier return.

The value of *n* remains in effect during a terminal session until changed by another LINESIZE command (or by an equivalent interpreter command).

At logon time, the assumed line length (in character positions) for valid terminals is as follows:

Terminal	Line Length
3101	72
3270	80
3767 SDLC	132
3767 S/S, 2741, 1050	120
3770	132
CPT-TWX	72

Physical margins are not affected by the `LINESIZE` command, nor is the allowable length of terminal entries affected.

Note: Care must be used in specifying the `LINESIZE` command. If the session line length specified is longer than the terminal's physical line length, and if lines of data are also longer than the physical line length, some of the terminal output will be lost during line display.

LIST Command

The `LIST` command displays all or part of the workspace contents.

LIST	<i>line</i> <i>line :end-line</i> *	Optional first operand—choose one or omit. If omitted, entire workspace displayed.
	LINE NOLINE	Optional—choose one or omit. If omitted, LINE is assumed.

line and *end-line*

specify integers that correspond to line numbers in the workspace. If the integers specified are not actual line numbers, existing lines within the specified range are displayed.

line

specifies that this line is to be displayed.

line :end-line

specifies that every line within the specified range of line numbers is to be displayed.

When both *line* and *end-line* are omitted, all lines in the workspace are displayed.

*

denotes the current line. It may be substituted for *line*, *end-line*, or both.

LINE or **NOLINE**

specifies the display convention to be used for line numbers.

LINE

specifies that line numbers are to be displayed.

NOLINE

specifies that line numbers are to be omitted.

When **LINE** and **NOLINE** are omitted, **LINE** is assumed.

When the `LIST` command is executed, the workspace is displayed according to the following rules:

- The workspace must not be empty.
- If the workspace contents are not owned by this user and have `PROTECT NOREAD` in effect, the workspace is not displayed and an error message is issued.

- If line numbers are being displayed, then a line that has been flagged as “deleted” by a foreground processor will be marked by an asterisk (*) immediately after the line number.

LOAD Command—to Retrieve VSPC Files

The LOAD command in this form places a specified file in the workspace.

LOAD	<i>filename</i>	Required operand.
-------------	-----------------	-------------------

filename

identifies the file to be placed in the workspace; it must conform to the rules for VSPC filenames (see the “Filenames” and “Passwords” entries).

The *filename* must identify a file that contains editable data. That is, the file may contain a command list, a data file, or a compiler source program. The file must not be an object program, an interpreter workspace, or an undefined file.

General Execution Considerations

When the LOAD command is executed, VSPC locates the specified file and reads it into the workspace. Any previous workspace contents are completely replaced. All workspace characteristics—such as name, attribute, access control options, and NUMBER conventions—are changed to match those of the new contents.

If the user issuing the LOAD command is not the owner of the specified file, and the file has PROTECT NOREAD in effect, then the workspace assumes the protected characteristic (see the “PROTECT Command” entry).

LOAD Job Entry Command—to Retrieve Output

The LOAD command in this form retrieves batch job output and makes it available to the terminal.

LOAD	OUTPUT(<i>jobname</i>)	Required operand.
	SMSG DS(<i>dataset-identifier</i>)	In OS/VS, optional operand; if omitted, SMSG is assumed.
	LINE(<i>begin-line</i>) LINE (<i>begin-line</i> : <i>end-line</i>)	Optional—choose one or omit. If omitted, first line of output is assumed.

OUTPUT(*jobname*)

tells VSPC that batch job output is to be placed in the workspace.

jobname

identifies the job whose output is to be retrieved. The *jobname* must conform to the rules for VSPC jobnames (see the “Jobnames” entry).

The entire *jobname* may be specified; alternatively, the *jobname* may be specified omitting the 6-character job entry code.

SMSG or DS(*dataset-identifier*)

specifies which data set in the VSPC class to retrieve. Only OS/VS systems use this operand.

SMSG

when specified, identifies the desired output as the system message data set.

When SMSG is specified, the batch system messages pertaining to this job are made available in the workspace.

DS(*dataset-identifier*)

when specified, is the uniquely identifying portion of the SYSOUT data set name; it must be enclosed in parentheses.

For OS/VS1 systems, *dataset-identifier* is the number of the DD statement for the data set as it appears in job control language, counting the SYSOUT=VSPC class DD statements only.

For OS/VS2 systems, *dataset-identifier* is that part of the dataset-name that uniquely defines the data set—in JES2, the last four digits of the key-number in the last qualifier; in JES3, the ddname, stepname.ddname, or procstepname.stepname.ddname. If fewer (or more) characters are specified, the system will select the first SYSOUT data set it finds whose name ends with the specified character string. In OS/VS2, *dataset-identifier* may be enclosed in single quotation marks within the parentheses and *must* be so enclosed if it contains any periods.

In either case, the *dataset-identifier* can be determined by inspecting the batch system messages for those with MSGCLASS=*vspc-output-class*, provided that the second subparameter of the MSGLEVEL parameter was specified as 1 when the job was submitted.

When both SMSG and DS(*dataset-identifier*) are omitted, SMSG is assumed.

LINE(*begin-line*) and LINE(*begin-line :end-line*)

specify that only part of the specified output is to be loaded.

begin-line and *end-line*, when specified, must be integers. *begin-line* identifies the first line to be loaded; *end-line* identifies the last line to be loaded. The *end-line* must be equal to or greater in value than *begin-line*.

LINE(*begin-line*)

when specified, only that part of the output is loaded that begins with the beginning line number and ends with the end of the output.

LINE(*begin-line :end-line*)

when specified, only that part of the output is loaded that begins with the beginning line number and continues through the ending line number, inclusive.

When the LINE operand is omitted, the entire specified output is loaded.

Whether the LINE operand is specified or omitted, if the entire workspace is filled before the specified data is entirely loaded, loading stops.

General Execution Considerations

In OS/VS, batch job output is available for retrieval only if the user specified the proper JCL output class parameter in the input job stream. This output class is defined by the system administrator; it is the only one accessed when the LOAD command is executed. In certain cases, such as when jobs are routed through the OS/VS2 MVS Network Job Entry Facility for execution, SYSOUT cannot be retrieved unless MSGCLASS also specifies the same special output class.

In OS/VS1, any output line longer than 256 characters is truncated. In OS/VS2, any output line longer than 4058 characters is truncated, and data sets containing records longer than 4096 bytes are not supported.

If loading stops because the entire workspace is filled, another LOAD command can be issued for the remaining output lines after the first group of lines is saved.

After the output data is loaded, the workspace has the DATA attribute and is unnamed. Any prior PROTECT, SHARE, or NUMBER characteristics are removed. Line numbers begin with *begin-line*, if specified, or 1; they are incremented by 1 for each logical record retrieved.

Local Commands—View Mode Only

Local commands, valid only for display terminals in View Mode, are entered over the line-number of a file. They specify an action to be taken at the line where they are entered. The general form of a local command is:

nnnnX

nnnn

is an optional number of up to four digits that specifies the number of lines to be acted upon. (For the / local command, this number has no meaning and is ignored.)

X

is a character that specifies a local command. The characters and their meanings are:

/—change current line pointer

specifies that this line becomes the current line.

If more than one / local command is specified upon one screen display, only the last specified is acted upon; previous / local commands on that screen are ignored.

For this command *nnnn* has no meaning; if it is specified, it is ignored.

A—add blank lines

specifies that *nnnn* blank lines are to be added to the workspace after this line.

When the A local command is executed, *nnnn* blank lines are inserted into the workspace after this line. Line numbers for the added lines begin at this line + 1 and the increment is one. When necessary, subsequent lines in the workspace are renumbered in increments of one as far as necessary to maintain sequential line numbering.

The blank lines have a length of one character, except for DIRECT files. DIRECT file blank lines have the same length as all other lines in the file.

D—delete lines

specifies that *nnnn* lines, beginning with this line, are to be deleted.

When the D local command is executed, *nnnn* lines, beginning with this line, are deleted from the file; if any local commands are specified within this range of lines, the commands are not executed.

R—repeat lines

specifies that this line is to be copied *nnnn* times in new lines generated immediately after this line.

Line numbers are generated in the same way as they are generated for the A local command.

General Execution Considerations

When a local command is entered, the line number in the workspace is not changed; after the local command is executed the line number (if the line still exists in the workspace) is restored when the screen is redisplayed.

Multiple local commands can be entered on the screen before the commands are executed (either through pressing the ENTER key or through a preset PF key); the commands are executed in the order they are specified from top to bottom on the screen, with the following special considerations:

- All D (delete) commands are processed in the order they appear on the screen before any others.
- All other commands are then processed in the order they appear on the screen.

LOCATE Command

The LOCATE command finds or relocates the current line pointer.

LOCATE	<i>line</i> + <i>number</i> - <i>number</i> <i>begin-line</i> 'string ' <i>begin-line</i> : <i>end-line</i> 'string ' 'string ' *	Optional first operand—specify one or omit. If omitted, find current line is assumed.
	NOTEXT TEXT	Optional second operand—ignored in View Mode. Otherwise, if omitted, NOTEXT is assumed (unless TEXT ON has been specified).

line

specifies an absolute line number to which the current line pointer is to be set.

+ *number* and - *number*

specify a relative number of lines by which the current line pointer is to be set higher (+ *number*) or lower (- *number*) in the workspace.

begin-line and *end-line*

specify absolute line numbers that indicate the beginning and ending of a range of numbers to be searched for 'string '.

If *begin-line* only is specified, the search begins at that line and continues to the end of the workspace.

If both *begin-line* and *end-line* are omitted, the search begins with the line after the current line and continues to the end of the workspace.

'string '

specifies a group of contiguous characters for which VSPC is to search the specified or implied range of lines; VSPC then resets the current line pointer to the first line found that contains the specified characters. If the string contains only alphabetic and numeric characters, then the quotation marks may be omitted. However, if quotation marks are omitted, lowercase characters in the string will be automatically translated to uppercase for command entry. To get a string containing lowercase characters, specify TRANSLATE OFF and enclose the string in quotation marks.

*

can be specified by any absolute line number The * denotes the current line.

NOTEXT

specifies that the VSPC response to the command should be the line number of the new current line.

TEXT

specifies that the VSPC response to the command should be the line number and text of the new current line.

If both TEXT and NOTEXT are omitted, NOTEXT is assumed, unless a TEXT ON Command has been specified (see TEXT Command entry).

General Execution Considerations

When the LOCATE command is executed, VSPC uses the information supplied in the operands to reset the current line pointer.

If the 'string' operand is specified, and the group of characters cannot be found, the current line pointer is unchanged and the following message is displayed:

LINE NOT FOUND

If an absolute line number is specified, and that line does not exist in the workspace, the current line pointer is set to the next lower line number, or (if no lower line number exists) to the first line in the workspace.

If a request for a relative line number exceeds the workspace limits, the current line pointer is set as follows:

- To the first line in the workspace if - is specified
- To the last line in the workspace if + is specified

When the LOCATE command is entered in View Mode, TEXT/NOTEXT is ignored but the workspace display area is changed, if necessary, to display and highlight the new current line.

When the LOCATE command is entered in Command Mode, the line number (NOTEXT) or the line number and text (TEXT) of the current line are displayed.

MERGE Command

The MERGE command combines a file with the contents of the workspace.

MERGE	<i>filename</i>	Required first operand.
	<i>destination-line</i> <i>destination-line</i> <i>increment</i> * OVERLAY	Optional—choose one or omit. If omitted, file placed at end of workspace contents.

filename

identifies the file to be retrieved; it must contain editable data; it must conform to the rules for VSPC filenames (see the "Filenames" and "Passwords" entries).

destination-line

identifies the line in the workspace after which the file is to be inserted. It must be an integer from 0 through 99999. If the *destination-line* specified is not an actual line number, the next lower existing line in the workspace is assumed.

increment

specifies the increment to be used for numbering the inserted file. It must be an integer from 1 through 99999. When *increment* is omitted, an increment of 10 is assumed.

*

denotes the current line. It may be substituted for *destination-line*.

OVERLAY

specifies that the file and the workspace are to be combined, based on their existing line numbers.

General Execution Considerations

When the MERGE command is executed, VSPC retrieves the file and combines it with the workspace contents, according to the following rules:

- If both *destination-line* and OVERLAY are omitted, the file is placed at the end of the workspace contents. The line-number given the first added line is the next multiple of 10 higher than any existing line-number. For the added lines, the increment is 10.
- When *destination-line* is specified, the file is inserted into the workspace as if it were the subject of a MOVE command (see the "MOVE Command" entry). If *increment* is specified, that increment is used for numbering the inserted lines; otherwise the increment is 10. If the last generated line-number overlaps the next existing line-number after the inserted lines, then the succeeding original lines are also renumbered—as far as is necessary to maintain the ascending sequence of line numbers—according to the previous rules. The following message is sent:

```
RENUMBERED FROM LINE nnn
```

where *nnn* is the number of the first original line to be renumbered.

- When OVERLAY is specified, the file and the workspace contents are combined according to the ascending sequence of their existing line numbers. If two line numbers are equal, the file line replaces the workspace line. (See the example given in Figure 9.)

The attribute of the workspace is not changed. If the inserted file has a different attribute, MERGE command execution is completed, and the following warning message is issued:

```
WARNING - DIFFERENT FILE TYPE MERGED
```

If the file has the PROTECT NOREAD characteristic, and is not owned by this user, then, after the MERGE command is executed, the workspace assumes the protected characteristic, and any commands (as, for example, LIST) that would reveal the contents of the workspace cannot be executed.

MESSAGE Command

The MESSAGE command specifies whether or not messages are to be displayed at the terminal and/or whether or not message headers are to be displayed.

MESSAGE	ID NOID	Optional—choose one or omit. If omitted, present display convention is unchanged.
	BLOCK OPEN WAIT	Optional—choose one or omit. If omitted, present message mode is unchanged.
	If all operands are omitted, current value is displayed.	

ID and NOID

specify the display convention for message headers. (A unique identifying header precedes each VSPC message; the header form is *ASU nnn* where *nnn* is a three-digit number.)

ID

specifies that VSPC message headers are to be displayed with each message.

NOID

specifies that VSPC message headers are to be omitted—only the message text is to be displayed.

When both are omitted, the current display convention for message headers is unchanged.

BLOCK, OPEN, and WAIT

specify the message mode in which the terminal is to operate.

BLOCK

prevents display of unsolicited messages at the terminal. VSPC responses to commands are not suppressed.

OPEN

specifies normal terminal operation—that is, messages can be received whenever the terminal is not open for input. Terminal input can be entered without a previous attention signal.

WAIT

specifies that the terminal is continuously ready to receive messages. In this mode, an attention signal must precede any input from the terminal.

When **BLOCK**, **OPEN**, and **WAIT** are omitted, the present message mode is unchanged.

At logon time, the **NOID** and **OPEN** operands are in effect.

Note: If the MESSAGE mode is changed from **BLOCK** to **OPEN**, the most recent general message from the VSPC operator (if any was sent after MESSAGE **BLOCK** went into effect) is displayed at the terminal.

MOVE Command

The MOVE command repositions one or more lines of editable data within the workspace.

MOVE	<i>line</i> <i>line:end-line</i> *	Required first operand—choose one.
	<i>destination-line</i> *	Required second operand.
	<i>increment</i>	Optional operand—specify or omit. If omitted, 10 is assumed.

line and *end-line*

must be integers within the range 0 through 99999.

line

specifies the present line number of a single line to be moved. The line number specified must actually exist in the workspace.

line :end-line

specifies the present position of a range of lines to be moved; *line* specifies the first line in the range, and *end-line* specifies the last. The line numbers specified need not actually exist in the workspace; if they do not, then existing lines within the range specified are moved. The *end-line* must be equal or greater in value than *line*.

destination-line

is an integer that specifies the line number after which the range of lines is to be positioned; it need not actually exist within the workspace. It may be either lower in value than *line* or greater in value than *end-line*. That is, it must not be within the range from *line* through *end-line*, inclusive.

*

denotes the current line. It may be substituted for any of the following: *line*, *end-line*, or *destination-line*.

increment

specifies the increment by which moved lines are to be renumbered. It must be an integer in the range 1 through 99999.

General Execution Considerations

The following execution rules apply:

- If *line* is specified and *end-line* is omitted, then the one corresponding line is moved to a position immediately following *destination-line*, if *destination-line* actually exists. If *destination-line* does not actually exist, the line is inserted using the specified destination line number.
- If *line* and *end-line* are both specified, then the inclusive range of corresponding lines is moved to a position immediately following *destination-line*, if *destination-line* actually exists. If it does not, they are inserted with the first moved line using the *destination-line* line number.

- Moved line(s) are renumbered sequentially higher than the line that corresponds to *destination-line*. If *increment* is specified, that increment is used in renumbering; if *increment* is omitted, the increment is 10.
- If the last generated line number overlaps the next existing line number after the inserted lines, then the succeeding original lines are also renumbered—as far as is necessary to maintain the ascending sequence of line numbers—according to the previous rule. The following message is sent to the user:

RENUMBERED FROM LINE *n*

where *n* is the number of the first original line to be renumbered.

When execution is successful, the following message is sent:

n LINES MOVED

where *n* is the number of lines that have been moved.

Note: If a MOVE command is executed for a workspace that corresponds to a direct file, the workspace will no longer meet direct file requirements, and, unless the file is renumbered using the RENUMBER command, will cause an error message when the user attempts to save it in a library.

NAME Command

The NAME command identifies the workspace.

NAME	<i>filename</i>	Optional operand—choose one or omit. If omitted, name of current workspace is displayed.
------	-----------------	--

filename

specifies the identifying name to be used; it must conform to the rules for VSPC filenames (see the “Filenames” and “Passwords” entries).

When the name field of the *filename* is CONTINUE, only the user’s own library number may be specified or implied.

General Execution Considerations

When the NAME command is executed, the specified *filename* replaces any previous filename identifying this workspace. When:

- the new *filename* —not including the optional password—is different from the workspace filename, and
- the library contains a nonempty file with the same name,

the following warning message is issued:

WARNING - FILE ALREADY EXISTS

If the workspace under the new *filename* is later placed in the library, it replaces the previously existing file with the same name. If this is not the desired result, a new NAME command can be issued, specifying a different name.

NEWLINE Command

The NEWLINE command specifies a character which, when entered into the workspace, is translated into a nonprintable new line character.

NEWLINE	'x' ' '	Optional operand—choose one or omit. If omitted, current value is displayed.
----------------	------------	--

'x'

specifies a character to be translated into the nonprintable new line character.

' '

(two adjacent quotation marks) specifies that no character is to be substituted for the new line character. (If a substitute character is currently specified, the specification is removed.)

General Execution Considerations

After the NEWLINE command is issued, one of the two following actions takes place.

- If a substitute new line character is specified, the specified character whenever encountered in input is translated into the nonprintable new line character when it is entered.
- If two adjacent quotation marks are specified, no substitute character for the new line character is recognized.

After a NEWLINE command specifying a substitute character is executed, a single command line can contain several VSPC commands; the commands are executed as if each were entered on a separate line.

If the NEWLINE character is entered in a command in a quoted string, then it is treated as a literal newline character not as a line separator.

In INPUT mode, the substitute new line character is treated as a line separator. That is, when the character is encountered upon input, the current line is ended at the character before the new line character; the next line number and a space are generated and are followed by the data following the new line character.

NUMBER Command

The NUMBER command specifies the sequence numbering convention to be used for a file during subsequent batch processing.

NUMBER	<i>filename</i> *	Optional first operand—choose one or omit. (Required if other operand is specified.)
	LEFT LEFT <i>n</i> RIGHT RIGHT <i>n</i> LIST LIST <i>n</i> NONE	Optional second operand—choose one or omit. (Required if other operand is specified.)
	If all operands are omitted, current values are displayed.	

filename or *

specifies the area to which the numbering convention is to apply.

filename

specifies the file for which the sequence numbering convention is to be used. The file must be owned by this user. The *filename* must conform to the rules for VSPC filenames (see the "Filenames" and "Passwords" entries).

filename must identify a SEQUENTIAL or DIRECT file, and must not specify the special file CONTINUE.

*

specifies that the sequence numbering convention is to be used for the contents of the workspace.

LEFT, RIGHT, LIST, or NONE

specifies the numbering convention to be used in generating sequence numbers. (The sequence numbers are generated from the existing VSPC line numbers.) One and only one of these operands must be specified.

LEFT

specifies that sequence numbers are to be placed in the leftmost six character positions; when necessary, leftmost zeros are used to fill out the six character positions. Line text begins at character position seven; if necessary, the line is padded with blanks to fill the output record.

LEFT *n*

specifies that sequence numbers are to be placed in the leftmost *n* character positions and if necessary padded with leftmost zeros to fill out the *n* character positions. *n* may be specified as an integer from 1 through 8, inclusive. Line text begins at character position *n* + 1; if necessary, the line is padded at the right with blanks to fill the output record.

RIGHT

specifies that sequence numbers are to be placed in the rightmost eight character positions in the record; when necessary, leftmost zeros are used to fill out the eight character positions. Line text begins at character position 1; if necessary, the line text is padded with blanks to fill out the remaining character positions.

RIGHT *n*

specifies that sequence numbers are to be placed in the rightmost *n* character positions, and if necessary, padded with leftmost zeros to fill out the *n* character positions. *n* may be specified as an integer from 1 through 8, inclusive. Line text begins at character position 1; if necessary, the line text is padded at the right with blanks to fill out the remaining character positions.

LIST

specifies that line numbers are to be placed in the leftmost five character positions; when necessary leftmost zeros are used to fill out the five character positions. A blank in character position six follows the line number, and line text begins at character position seven. If necessary, the line text is padded at the right with blanks to fill the output record.

LIST *n*

specifies that line numbers are to be placed in the leftmost *n* character positions; when necessary leftmost zeros are used to fill out the *n* character positions; *n* may be specified as an integer from one through eight. A blank in character position *n*+1 immediately follows the line number. Line text begins at character position *n*+2. When necessary, the line text is padded at the right with blanks to fill the output record.

NONE

specifies that line numbers are not to be converted to sequence numbers. Line text is placed in character position one. When necessary, line text is padded at the right with blanks to fill the output record.

General Execution Considerations

During batch job sequence number generation by the SUBMIT command, if a generated number exceeds the specified number of characters, record formatting stops, and an error message is issued. Also, the sequence number length plus the record length must not exceed 80 characters.

When NUMBER * is issued, the specified numbering convention remains in effect for the workspace until the next NUMBER *, LOAD, ENTER, or CLEAR command is executed. If the workspace contents are placed in a library, this sequence numbering convention is in effect for the resulting file.

When the NUMBER command is not specified for a workspace or file used in batch processing, the following options are implied:

- For any BASIC language attribute, NUMBER LIST is assumed.
- For all other attributes, NUMBER NONE is assumed.

These implied sequence numbering conventions are set by the ENTER and CLEAR commands.

Note: The NUMBER command sequence numbering convention is used by the SUBMIT command (see the "SUBMIT Command" entry) and also by certain VSPC Service Program control statements (check with your system administrator on their use).

OFF Command

The OFF command ends a terminal session.

OFF	CONTINUE	Optional—specify or omit. If omitted, CONTINUE file not saved.
------------	-----------------	--

CONTINUE

specifies that the current work is to be saved. If the workspace is password protected, the work is saved with this password.

When CONTINUE is omitted, the current work, and any existing CONTINUE file, is deleted when the OFF command is executed.

General Execution Considerations

When the OFF command is successfully executed, the current terminal session is ended, and the logoff messages are printed.

Unless VSPC is closing down, another logon request can be entered.

If a new CONTINUE file is not being saved, the OFF command causes any previous CONTINUE file in the user's library to be automatically purged. If CONTINUE is specified, the previous CONTINUE file (if any) is purged when the new CONTINUE file is successfully saved.

However, if the new CONTINUE file cannot be saved (for example, if the user's workspace is empty), any previous CONTINUE file remains in the user's library and the following message is sent to the user:

```
ASU228  UNABLE TO SAVE 'CONTINUE'
```

If an OFF command is issued from a 3270 with less than four unused lines in the output area, some of the logoff messages are lost when the CLEAR key is pressed. If the logoff messages are to be examined in their entirety, the screen should be cleared before the OFF is issued.

PASSWORD Command

The PASSWORD command supplies a new logon password.

PASSWORD	<i>password</i>	Optional—specify or omit. If omitted, password prompt is issued.
-----------------	-----------------	--

password

replaces the existing logon password. The *password* must comply with the VSPC rules for passwords (see the "Passwords" entry).

General Execution Considerations

When the PASSWORD command is executed, the new logon *password* replaces the current logon password in the user profile. The new logon password will be required to begin the next session.

The PASSWORD command can be executed any time during a terminal session.

For security purposes, the **PASSWORD** command can be entered without specifying the replacement *password*. VSPC responds with the following message:

```
ENTER PASSWORD
```

The user can then enter the new password without displaying it legibly at the terminal. If the user enters a null line (carrier return) instead of a new password, the requirement to supply a password at logon is removed.

If the user's profile does not specify a password, the **PASSWORD** command can be used to create one.

Passwords

Passwords allow increased VSPC programming and file security.

A *password* contains from one through eight alphabetic capital letters or numeric characters in any combination.

During terminal entry of passwords, translation of lowercase alphabetic characters to capital letters is always performed, even if **TRANSLATE OFF** is in effect.

In VSPC, *passwords* can be defined for logon and for files.

Logon passwords are initially created by the VSPC administrators. The user can change them with the **PASSWORD** command.

File passwords can be specified using the **NAME**, **SAVE**, **STORE**, **FILE**, and **PROTECT** commands. When they are so specified, they are required for:

- Library access using the **LOAD**, **MERGE**, **RUN**, and **SUBMIT** commands
- Directory entry modification using the **FILE**, **NUMBER**, **PROTECT**, and **SHARE** commands

When a *password* is specified in any of the above cases, it must be correctly supplied from the terminal before the intended processing can begin.

If no password is defined for a file, but the user supplies one when accessing the file, VSPC treats it as an incorrect password.

For increased security, passwords are never displayed as part of a profile or directory listing.

Password Assignment

The user can assign passwords to files through the **NAME**, **SAVE**, **STORE**, **FILE**, and **PROTECT** commands. In entering the command, the user can specify the fully qualified filename, including the password.

However, to suppress display of the password at the terminal, the user can specify the filename operand in the following form:

```
[libnum]name/
```

VSPC then issues prompting messages for the password as described in the following paragraphs.

Password Prompting

VSPC prompts for a password by displaying one of the following messages:

```
ENTER PASSWORD  
ENTER PASSWORD FOR filename  
ENTER NEW PASSWORD FOR filename
```

The user then enters the password; however, it is never legibly displayed at the terminal.

During logon, ENTER PASSWORD is always displayed, if a logon password is included in the user's profile (as described in the "VSPC Command" entry).

In other commands, when the required file password is not supplied in the appropriate operand, a prompting message is displayed. It is issued when a file password is being created or changed or when a specific matching file password must be supplied.

Before the command can be executed, the valid password must be entered from the terminal.

If an incorrect matching password is entered, VSPC issues the following message:

```
INCORRECT PASSWORD - RE-ENTER
```

If the valid password is not supplied within three reprompting attempts, VSPC prints the following message:

```
PASSWORD ERROR
```

The command is then rejected. (See also the "VSPC Command" entry.)

If a syntactically invalid new or changed password is entered, only one attempt is allowed, and VSPC issues the following message:

```
PASSWORD ERROR
```

The command is then rejected.

If CLIST processing is being executed, the processing terminates. (Foreground processors take other actions; see the appropriate terminal user's guide for the actions taken.)

If a carrier return is entered when VSPC prompts for a matching password, then VSPC responds as if all three prompting sequences have been executed.

If a carrier return is entered when VSPC prompts for a new or changed password, then the password is removed. During subsequent processing, no password is needed.

PFKEY Command

The PFKEY command specifies the interpretation of a program function (PF) key on the 3270 terminal.

PFKEY	<i>n</i>	Optional first operand—specify or omit. (Required if other operand is specified.)
	' ' 'string ' 'string ' ENTER 'string ' NOENTER 'string ' COMMAND ENTER 'string ' COMMAND NOENTER COMMAND COMMAND ENTER COMMAND NOENTE LAST LAST ENTER LAST NOENTER HCPY FORWARD(<i>x</i>) BACKWARD(<i>x</i>)	Optional second operand—choose one or omit. (Required if other operand is specified.)
	If all operands are omitted, current values are displayed.	

n
 specifies the number of the PF key to which you want to assign the interpretation. It must be between 1 and 24 inclusive.

' '
 this operand (two adjacent single quotation marks) specifies that any currently specified PF key definition be removed.

'string '
 specifies a string of characters to be inserted into the screen image at the location of the cursor when the selected PF key is pressed. (The action is as if the INS MODE key is pressed, the characters typed on the keyboard, and then the RESET key is pressed.) The cursor is relocated at the character position following the inserted string.

The maximum length of a string is 255 characters. The maximum total length of character strings assigned to PF keys is 600 characters.

COMMAND

specifies that the cursor is to be logically moved to the beginning of the command area before performing the function specified for the key. If COMMAND is not specified, the function (such as string insertion) is performed at the current location of the cursor on the screen.

LAST

specifies that, when the PF key is pressed, the contents of the last previous terminal input line are inserted into the command area at the current location of the cursor. The cursor is repositioned following the inserted characters.

ENTER

specifies that the string be copied to the screen as described above, and then the appropriate action taken as if the ENTER key had been pressed.

NOENTER

specifies that the string be copied to the screen but no other action taken. You can then modify it before pressing ENTER. Your line is truncated if the insert extends beyond the current input field.

If neither ENTER nor NOENTER is specified, ENTER is assumed.

HCPY

specifies that subsequent use of the named PF key should simulate selecting the HCPY field on the screen through cursor positioning and the ENTER key.

FORWARD(x) and BACKWARD(x)

specify that subsequent use of the selected PF key page the display FORWARD or BACKWARD *x* pages, as if the light pen or cursor positioning were used to select one of the forward/backward page control fields.

The maximum value the selected PF key can page is ninety-nine.

PROTECT Command

The PROTECT command changes access to a currently existing file or to one that may be created later.

PROTECT	<i>filename</i> *	Optional first operand—choose one or omit. (Required if other operands are specified.)
	PASSWORD PASSWORD(<i>password</i>) NOPASS	Optional—choose one or omit. If omitted and first operand is specified, password protection unchanged.
	READ NOREAD	Optional—choose one or omit. If omitted and first operand is specified, access characteristic unchanged.
	WRITE NOWRITE	Optional—choose one or omit. If omitted and first operand is specified, access characteristic unchanged.
	If all operands are omitted, the value applying to the current workspace is displayed.	

filename or *

specifies the VSPC area to which the command will apply. One is required as the first operand.

filename

specifies the file to which the PROTECT characteristics will apply; *filename* must conform to the VSPC rules for filenames (see the "Filenames" and "Passwords" entries).

*

specifies that the workspace contents are to have the PROTECT characteristics. The PROTECT characteristics go into effect when the workspace is placed in a library as a file.

PASSWORD, PASSWORD(*password*) or NOPASS

specifies that the current password protection is to be changed.

PASSWORD or PASSWORD(*password*)

supplies a new *password* for the file. If the new *password* is omitted from the command specification, VSPC prompts the user to supply one (see the "Passwords" entry).

NOPASS

removes an existing password from the file.

READ and NOREAD

specify read characteristics for users other than the owner.

READ

allows users other than the owner to use the file as input for an executing program or to include it in a SUBMIT job entry command. READ also changes a previous NOREAD characteristic.

NOREAD

prevents users other than the owner from using the file as input for an executing program, or including it in a SUBMIT job entry command. It also changes a previous READ characteristic. Another user may, however, specify this file in a LOAD, MERGE, or RUN command, if he has access to it. The NOREAD operand may not be specified for an object program file.

When a LOAD, RUN, or MERGE command places a file with NOREAD characteristics in the workspace, the entire workspace assumes the PROTECT characteristic. This means that the following commands are rejected: SAVE, STORE, CHANGE, FIND, and LIST.

WRITE and NOWRITE

specify write characteristics for the owner.

WRITE

specifies that the file's owner may modify this file.

NOWRITE

protects the file from modification, even by its owner. The file cannot be used for output or update by an executing program, nor can it be named in a SAVE, STORE, or PURGE command.

General Execution Considerations

Execution of the PROTECT command changes access to the file. Only the owner of the file may specify the PROTECT command with one exception: A library manager may specify PROTECT WRITE for a file in his library that he does not own.

When the PROTECT command is issued, at least one of the optional operands must be specified.

The PROTECT command cannot be issued for the special file CONTINUE.

A cleared workspace has the READ and WRITE characteristics; these may be explicitly changed through the PROTECT command; they can also be implicitly changed by loading a file which has the NOREAD or NOWRITE

characteristics. A newly created file has the **READ** and **WRITE** characteristics, unless it was created by saving a workspace with the **NOREAD** or **NOWRITE** characteristics.

If **WRITE** is the only keyword operand, and a password for the file exists, the password can be omitted when specifying the **PROTECT** command. However, if the password is specified, it must be specified correctly, or the command is rejected.

For the effect of the **PROTECT** command when specified with the **SHARE** command, see Figure 17.

PUNCH Command

For **CPT-TWX** terminals, the **PUNCH** command specifies whether or not terminal output will be punched on paper tape.

PUNCH	ON OFF CONTROL	Optional operand—choose one or omit. If omitted, current value is displayed.
--------------	---	--

ON and OFF

specify whether terminal output will be punched on paper tape.

ON

specifies that terminal output will be punched on paper tape. When **PUNCH ON** is in effect, **VSPC** adds the characters:

XOFF RUBOUT RUBOUT RUBOUT

at the end of each output line.

OFF

specifies that terminal output will not be punched on paper tape. When **PUNCH OFF** is in effect, no characters are added at the end of a line.

At logon, **PUNCH OFF** is in effect. Whether or not punched output is being produced, output is printed at the terminal as well.

CONTROL

specifies that data sent to the terminal or the attached paper tape punch contains all ASCII characters, including control characters, and that the data should not be edited by **VSPC**.

Whether or not terminal output is punched on paper tape depends on the manual setup of the terminal. If the setup is wrong, punched output is either incorrectly formatted or nonexistent.

PURGE Command

The PURGE command removes a file from a library.

PURGE	<i>filename</i>	Required operand.
--------------	-----------------	-------------------

filename

identifies the file to be removed. It must conform to the rules for VSPC filenames (see the "Filenames" entry).

If the file has a password, the password can be omitted when the PURGE command is issued. If, however, the password is specified, it must be specified correctly or the command is rejected.

General Execution Considerations

When the PURGE command is executed, the specified file is physically removed from the library.

For any file, the file owner may validly specify the PURGE command. For files in public or project libraries, the library manager may also specify the command. For other users, the PURGE command is rejected.

If PROTECT NOWRITE is in effect for this file, the PURGE command is rejected.

QUERY Command—to Display File Status

This form of the QUERY command displays status information about a file.

QUERY	FILE(<i>filename</i>)	Required operand.
--------------	------------------------------	-------------------

FILE(*filename*)

specifies that selected information about the named file is to be displayed.

filename

specifies the selected file; *filename* must conform to the rules for VSPC filenames (see the "Filenames" entry). The file password, if any, is not required; if it is specified, however, it is checked.

General Execution Considerations

When this form of the QUERY command is executed, the following information is displayed:

OWNER: *usernum*—where *usernum* is the user number of this file's owner (displayed only if the file resides in a noncontrolled PROJECT or PUBLIC library).

FILE TYPE: *type* **CONTENT:** *content*—where *type* can be SEQUENTIAL, DIRECT, OBJECT, UNDEFINED SEQUENTIAL, or UNDEFINED DIRECT, and where *content* can be CLIST, DATA, or *processor-name*.

FILE SIZE: *m* **LIMIT:** *n*—where *m* is the present file size in direct access storage, expressed in characters, and where *n* is the maximum allowable size, expressed in characters.

NUMBER OF RECORDS: *n*—where *n* is the number of logical records in the file (omitted if the file type is OBJECT).

RECORD SIZE: *n*—where *n* is the length of the logical records in the file (displayed only if the file has the specifications required for a **DIRECT** or **UNDEFINED DIRECT** file, as explained in the “File Organization” entry, and the file is not empty).

SEQUENCE NUMBERS: *option*—where *option* can be one of the following:

Option	Explanation
LIST <i>n</i>	left margin, reserving <i>n</i> + 1 characters for a sequence number followed by a blank
LEFT <i>n</i>	left margin, reserving <i>n</i> number of characters
RIGHT <i>n</i>	right margin, reserving <i>n</i> number of characters

(Omitted if the **NUMBER** attribute for the file is **NONE**.)

WORKSPACE LENGTH: *n*—where *n* is the length of object program or interpreter workspace when it is running or loaded in main storage (displayed only if the file type is **OBJECT**).

ACCESS: *access-option(s)*—where *access-option(s)* can be **NOREAD**, **NOWRITE**, and/or **SHARE** (displayed only if at least one of these options is in effect for the file).

DATE LAST WRITTEN: *mm/dd/yy*—where *mm/dd/yy* is the date the file was last opened for output or update, or a **SAVE** or **STORE** command was issued for it, or the file was created by a **FILE** command.

DATE LAST READ: *mm/dd/yy*—where *mm/dd/yy* is the date the file was last opened for input or update, or a **LOAD** or **RUN** command was issued for it. (Omitted if the file has not been opened for input or update or loaded or run since it was last created or replaced.)

Note: Depending on the startup option chosen, the date field may be either *mm/dd/yy* (month/day/year) or *dd/mm/yy* (day/month/year).

QUERY Command—to Display Library Directory

This form of the QUERY command displays a list of the files in a library.

QUERY	LIBRARY	Required first operand.
	LIBRARY (<i>libnum</i>)	If <i>libnum</i> omitted, user's own library assumed.
	OWN	Optional—specify or omit. If omitted, files of all owners assumed.
	CONTENT (CLIST) CONTENT (DATA) CONTENT (<i>processor-name</i>)	Optional—choose one or omit. If omitted, files of all attributes assumed.
	TYPE (SEQUENTIAL) TYPE (DIRECT) TYPE (OBJECT) TYPE (UNDEFINED) TYPE (UNDEFINED SEQUENTIAL) TYPE (SEQUENTIAL UNDEFINED) TYPE (UNDEFINED DIRECT) TYPE (DIRECT UNDEFINED)	Optional—choose one or omit. If omitted, all file types assumed.
	FROM (<i>name</i>)	Optional—specify or omit. If omitted, first file in library assumed.

LIBRARY

specifies that a list of the files in a library is to be displayed.

libnum

when specified, identifies the library to be listed. If *libnum* is omitted, the user's library is assumed. When *libnum* is specified, it must identify a library to which this user normally has access. If a *libnum* of zero (0) is specified, this user's project library is assumed.

OWN

when specified, requests that only names of files which belong to this user within the specified library be displayed.

When OWN is specified for a project or public library, only the user's own files are included in the list. OWN has no meaning, and is ignored, when specified for a private library.

CONTENT

when specified, requests that only names of files of the specified parenthesized attribute be displayed.

When **CONTENT** is specified, only names of files with the specified attribute (**CLIST**, **DATA**, or *processor-name*) are displayed. The *processor-name*, when specified, must be valid for this VSPC installation. It may be an IBM program product processor (as shown in Figure 15), or a user-written compiler or interpreter.

TYPE

when specified, requests that only files of the specified parenthesized type be displayed.

When **TYPE** is specified, only files of the specified type (**SEQUENTIAL**, **DIRECT**, **OBJECT**, **UNDEFINED SEQUENTIAL**, or **UNDEFINED DIRECT**) are included in the list. Note that **UNDEFINED**, **UNDEFINED SEQUENTIAL**, and **SEQUENTIAL UNDEFINED** have identical meanings, and that **UNDEFINED DIRECT** and **DIRECT UNDEFINED** have identical meanings.

FROM

when specified, identifies the file *name* at which the list is to begin.

Display begins with the first name which is equal to or higher than the specified *name*, and continues through the end of the library in ascending order according to the EBCDIC collating sequence—that is, in alphameric order as follows: \$, #, @, A, ... Z, 0, 1, ... 9. (The EBCDIC collating sequence is given in *System/370 Reference Summary*.)

Canceling output during directory display causes the operation to end.

QUERY Command—to Display User's Status

The QUERY command displays selected information about this user's VSPC status.

QUERY	TABSET TRANSLATE ACCOUNT PROFILE WORKSPACE	Optional—choose one or omit. If omitted, WORKSPACE is assumed.
--------------	---	---

TABSET

requests that the current session tab settings be displayed. If a substitute tab character is currently in effect, it is displayed; the EXPAND or NOEXPAND option is also displayed.

TRANSLATE

requests that the current translation convention (*tablename*, CAPS, or OFF) be displayed.

ACCOUNT

requests that accounting information be displayed.

Depending on the method your organization chooses, the information may be displayed as separate counts, as a composite accounting unit, or as a combination of the two. If separate counts have been selected, the following information is displayed:

CONNECT TIME: *hh:mm:ss*—where *hh:mm:ss* (hours:minutes:seconds) is the clock time (total elapsed time) this user has been logged on to VSPC.

HARDCOPY PAGES: *n*—where *n* is the number of hardcopy pages that have been printed on a printer attached to a 3270 terminal. (Omitted if none have been printed.)

TERMINAL ACCESSES: *n*—where *n* is the number of times this terminal user has communicated with VSPC (each communication being the execution of a command, a data entry, or a computer response).

CPU TIME: *nnn.nn SECS*—where *nnn.nn* is the total amount of time in seconds this user has utilized the central processing unit.

WORKSPACE USAGE: *n*—*n* expresses the utilization of virtual storage by this user's workspace, where *n* is thousands of bytes times seconds.

SPACE IN USER'S LIBRARY: *n*—where *n* is the total size of this user's files in his own library, in characters. That is, if this user is the library manager of a noncontrolled project or public library, only his own files are counted into the size.

SPACE IN OTHER LIBRARIES: *n*—where *n* is the total size of this user's files, in characters, in other libraries to which this user has access.

FILE ACCESSES: *nnn*—where *nnn* is the number of direct access input or output requests executed for this user (using either VSPC libraries or external VSAM files).

For example:

CONNECT TIME: 27:14:33
HARDCOPY PAGES: 1563
CPU TIME: 2034.71 SECS
SPACE IN USER'S LIBRARY: 12,769,201
SPACE IN OTHER LIBRARIES: 1,207,400
FILE ACCESSES: 71,367
TERMINAL ACCESSES: 135,203
WORKSPACE USAGE: 27,931,422

Note: The value in each field is the sum of the “to date” statistics for this user, plus the “current session” statistics.

If your organization has chosen to report composite units, one or more of the counts for CONNECT TIME, HARDCOPY PAGES, CPU TIME, FILE ACCESSES, TERMINAL ACCESSES, or WORKSPACE USAGE is omitted, and the following line appears:

PROCESSING nn.nn UNITS

where *nn.nn* is expressed in the composite units defined by your organization.

PROFILE

requests that this user's profile be displayed, as follows:

USERNUM: *usernum*—is the user number of this user.

LIBTYPE: *type*—specifies whether the user's library is PRIVATE, PROJECT, or PUBLIC (see the “Libraries” entry).

CONTENT: *attribute*—specifies this user's default workspace attribute (which may be different from that of the current workspace). The *attribute* can be DATA, CLIST, or (if one of the IBM program product foreground processors is installed, or a user-written compiler or interpreter is available) *processor-name*.

CPUMAX: *nnn* SECONDS—specifies, in seconds, the maximum CPU (central processing unit) time that can elapse between interactions between a foreground processor and the user. The default is zero, which indicates that there is no limit. Omitted if CPUMAX is zero.

SPACE: *nnn*—specifies the maximum number of characters this user's library can occupy.

SIZE *mmm* EDITABLE, *nnn* OBJECT—specifies the maximum workspace sizes as follows: *mmm* characters for editable data (DATA or CLIST or compiler source programs), *nnn* characters for object programs.

INTPRMAX: *mmm* MAXIMUM, *nnn* DEFAULT—specifies the maximum size (*mmm* characters) and default size (*nnn* characters) for an interpreter workspace.

SSMAX: *mmm* CHARS, *nnn* ITEMS—specifies the maximum size (*mmm* characters) for a single data item in shared storage, and the maximum number (*nnn*) of items the user may make known to the shared storage manager at one time.

JECODE: *code*—when this entry is present, the user is allowed to submit jobs for batch processing through the job entry commands. It specifies the first six characters of the jobname that identifies this user to the VSPC job entry facility. Omitted if this user does not have job entry privileges.

PRIVILEGE: *privilege*—specifies whether or not this user may use the privileged VSPC commands (not described in this manual). ADM and ACTADM specify that the user is allowed to use those commands. Omitted if this user does not have a privileged status.

PROJLIB (*libnum*)—specifies the project library to which this user has access. Omitted if this user does not have access to a project library.

ACCOUNT: '*string*' or **ACCOUNT:** '*string*' **JOB**—specifies up to 44 characters of accounting identification an organization assigns a user. The accounting identification is used to record the user's VSPC logons, and is optionally used when the user submits batch jobs. **JOB** is displayed if the accounting string is to be inserted in the accounting field of any **JOB** statements in remote jobs submitted by this user. This information is omitted if no accounting identification has been assigned to this user.

For example:

```
USERNUM: 12345
LIBTYPE: PRIVATE
CONTENT: DATA
CPUMAX: 5 SECONDS
SPACE: 2,500,000 BYTES
SIZE: 40,000 EDITABLE, 80,000 OBJECT
INTPRMAX: 60,000 MAXIMUM, 30,000 DEFAULT
SSMAX: 4096 CHARS 10 ITEMS
JECODE: DEPT87
PROJLIB: 22222
ACCOUNT: UNP12345
```

Note: If the VSPC administrator alters a user's profile after the user logs on, the **QUERY PROFILE** command displays the altered values; these values, however, do not take effect until the next time the user logs on.

WORKSPACE

requests that information about the workspace be displayed as follows:

NAME *filename*—the filename currently identifying the workspace. The *libnum* field in *filename* is displayed only if it is different from the user's library number (see the "Filenames" entry).

CONTENT *attribute*—the current attribute of the workspace contents.

LENGTH *nnn* **CHARS**—the number of data characters contained in the workspace.

nnn **LINES**—the number of lines contained in the workspace.

HIGHEST LINE NUMBER *n*—the line number of the last line in the workspace.

For example:

NAME: 3400 PROGLIST
CONTENT: CLIST
LENGTH: 922 CHARS, 50 LINES
HIGHEST LINE NUMBER: 500

When the QUERY command is executed, the requested information is displayed at the terminal in the format indicated.

RELEASE Command

The RELEASE command makes a file in a noncontrolled project library available for transfer of ownership.

RELEASE	<i>filename</i>	Required operand
----------------	-----------------	------------------

filename

identifies the file to be made available for transfer; it must conform to the rules for VSPC filenames (see the "Filenames" and "Passwords" entries). The file must be a noncontrolled project library file. If the file has a password, the password can be omitted when the RELEASE command is issued. However, if the password is specified, it must be specified correctly, or the RELEASE command is rejected.

General Execution Considerations

The RELEASE command can be issued only by the owner of the file or by the project library manager. However, files owned by the library manager are not eligible for ownership transfer; if the RELEASE command is issued for such files, it is rejected.

When the RELEASE command is executed, the specified noncontrolled project library file is made available for transfer of ownership.

This user retains ownership of the file until another user executes an ACQUIRE command naming this file, or executes a program which accesses the file for output or update. If the ownership is transferred explicitly by use of the ACQUIRE command, the released status is removed; if the ownership is transferred implicitly by accessing the file for output or update, the file remains released.

When ownership is transferred, the space this file occupies is subtracted from this user's file space and added to the new owner's.

RENUMBER Command

The RENUMBER command changes the line numbers of all or part of the workspace.

RENUMBER	<i>old-line</i> <i>old-line new-line</i> <i>old-line new-line</i> <i>increment</i> *	Optional—choose one or omit. If omitted, all lines renumbered with increment of 10.
-----------------	--	---

old-line

identifies a line number in the workspace. It must be an integer with a value from 0 through 99999.

new-line

when specified, must be an integer with a value from 0 through 99999. When specified, *new-line* represents the new value to be assigned to *old-line*.

*

denotes the current line. * may be substituted for *old-line*, *new-line*, or both.

increment

when specified, must be an integer that represents the increment used in assigning new line numbers; *increment* may have a value from 1 through 99999, inclusive.

General Execution Considerations

When the RENUMBER command is executed, line numbers in the workspace are reassigned. The workspace must not be empty. The following rules apply:

- If all operands are omitted, all lines of data in the workspace are renumbered. The first line is assigned line number 10; succeeding line numbers are incremented by 10.
- If only *old-line* is specified, the editable data beginning at *old-line* and continuing to the end of the data are renumbered. The new line numbers are multiples of 10, and are incremented by 10.
- If *old-line* and *new-line* are specified, then *old-line* (the old line number) is replaced by *new-line* (the new line number), and succeeding line numbers are incremented by 10. The new line number must be higher in value than the last line number preceding the old line number; that is, there must be no overlapping of old and new line numbers.
- If the value of *old-line* falls between the values of existing line numbers in the workspace, the next higher existing line number is the first line renumbered.
- If *increment* is specified, then the new line numbers are incremented by the value specified in *increment*.

If the workspace attribute is that of a BASIC language processor, internal line references in the BASIC source program are adjusted to the new line numbers during renumbering. A line number following the last statement of the renumbered program is reserved as a reference point for any statements that originally referred to nonexistent lines.

When execution is successful, VSPC responds with the following message:

n LINES RENUMBERED

where *n* specifies the number of lines renumbered.

Note: If a RENUMBER command is issued for a workspace that corresponds to a direct file, and any *increment* other than 1 is used, the workspace will no longer meet VSPC requirements for a direct file, and an error will occur when an attempt is made to place it in a library. For correct results, the RENUMBER command must be specified as:

RENUMBER 0 1 1

ROUTE Job Entry Command

The ROUTE command specifies an alternate destination for VSPC batch job output.

ROUTE	<i>jobname</i>	Required first operand.
	CLASS(<i>i</i>)	Optional—specify or omit. If omitted, predefined class assumed.
	DEST(<i>destination</i>)	Optional—specify or omit. If omitted, predefined destination assumed.
	SMSG DS(<i>dataset-identifier</i>)	Optional—choose one or omit. If omitted, entire job output assumed.

jobname

identifies the job to be routed. The *jobname* must conform to the rules for VSPC jobnames (see the “Jobnames” entry). The entire *jobname* may be specified; alternatively, the *jobname* may be specified omitting the 6-character job entry code.

CLASS(*i*)

when specified, *i* represents an alternate class to which the output will be routed.

If the CLASS operand is omitted, the output is routed to the default class for VSPC batch output, as defined by the VSPC administrator.

DEST(*destination*)

when specified, *destination* identifies a remote work station to which output will be routed, as defined by the system.

SMSG or DS(*dataset-identifier*)

specifies which data set in the job to reroute.

SMSG specifies the system messages data set.

DS(*dataset-identifier*) specifies the data set for which *dataset-identifier* is the uniquely identifying portion of the SYSOUT data set name (see the "LOAD Job Entry Command—to Retrieve Output" entry for instructions on determining the proper *dataset-identifier*).

General Execution Considerations

When the ROUTE command is executed, the batch system queues the job identified by *jobname* to the desired class, either explicitly as specified by the CLASS operand, or implicitly to the default VSPC class. If the DEST operand is specified, output is routed to that remote destination; otherwise it is routed to the normal destination. If the DS operand is specified (OS/VS only), only the specified data set is rerouted; otherwise the entire job is rerouted.

Upon successful execution, when the CLASS operand is omitted, VSPC returns the following response:

OUTPUT ROUTED TO CLASS *i*

where *i* represents the alternate output class, as defined by the organization or specified by the user.

RUN Command

The RUN command compiles and/or executes a program or command list.

RUN	<i>filename</i>	Optional—specify or omit. If omitted, workspace is assumed.
	'string'	Optional

filename

when specified, identifies the file to be processed. It must conform to the VSPC rules for filenames (see the "Filenames" and "Passwords" entries).

When *filename* is specified, the RUN command loads the specified file and operates upon that workspace. The attribute of the file must be CLIST or a compiler *processor-name*.

When *filename* is omitted, the RUN command operates upon the contents of the workspace. In this case, the attribute of the workspace must be CLIST or a compiler *processor-name*, as shown in Figure 15.

'string'

specifies a character string to be passed to the program. It may consist of 0 through 255 characters enclosed in single quotation marks. All characters are valid. However, if a single quotation mark (') is contained within the string, it must be specified as two adjacent single quotation marks (").

General Execution Considerations

When the RUN command is executed, one of the following actions occurs:

- If *filename* is omitted, the compiler source program in the workspace is compiled and executed, or the command list is executed.
- If *filename* specifies a compiler source program, the program is loaded, compiled, and executed.
- If *filename* specifies a compiler object program, the current data in the workspace is erased and the program is executed.
- If *filename* specifies a command list, the list is loaded and executed.

When the RUN command successfully operates upon a compiler program, the program is compiled and/or executed. A program header is printed, followed by any compiler messages, followed by terminal interactions with the executing program (input and output).

When the RUN command successfully executes a command list, VSPC executes (and prints or suppresses printing of) the commands in the list. A command list may cause another command list to be processed (through the RUN command); when the second command list completes processing, control is returned to the first command list. If a subordinate command list ends abnormally, then command list execution stops and all the connected command lists are considered to have ended abnormally. (See the "Processing Command Lists" section.)

If a string is specified in the command, the use of that string is processor dependent.

SAVE Command

The SAVE command causes the workspace contents to be placed in the specified file.

SAVE	<i>filename</i>	Optional—specify or omit. If omitted, current workspace name is used.
------	-----------------	---

filename

when specified, creates or changes the *filename* for the saved file. The *filename* must conform to the rules for VSPC filenames (see the "Filenames" and "Passwords" entries).

General Execution Considerations

When *filename* is specified, the workspace contents are placed in the file specified by *filename*. This *filename* will now identify the workspace (replacing any previous identifying *filename*).

If the SAVE command creates a new file, it may be placed in the user's private or project library, or any public library (except when the file is CONTINUE).

Only a file's owner may use the SAVE command to replace an existing file. The file type of the workspace and of the existing file must be the same

(except when the file is CONTINUE). The file must not have the NOWRITE protection characteristic.

If the *filename*—not including the optional password—is different from the workspace name and the library specified already contains a nonempty file with the same name, SAVE command execution is unsuccessful, and an error message is printed. This prevents accidental overwriting of an existing file by a new file.

When *filename* is omitted, workspace contents are placed in the file, using the *filename* of the workspace. No check for duplicate names is made, because VSPC assumes one of the following:

- The workspace contents were loaded from this *filename*, and the SAVE command simply updates the old file.
- This *filename* was specified with the NAME command, and at that time a warning message was issued which the user wishes to ignore (see the “NAME Command” entry).

When *filename* is omitted and the workspace is unnamed, an error message is issued, and execution is unsuccessful.

When CONTINUE is the name field for this file, the user’s own libnum must be specified or implied. The SAVE command overlays an existing CONTINUE file, even if CONTINUE is explicitly specified, and even if the file types are different.

When the library file is a direct file and the workspace does not qualify as a direct file (see the “File Organization” entry), an error message is issued, and execution is unsuccessful.

SCRATCH Job Entry Command

The SCRATCH command causes all or part of a job’s output to be deleted.

SCRATCH	<i>jobname</i>	Required first operand.
	SMSG DS(<i>dataset-identifier</i>)	Optional—choose one or omit. If omitted, entire job output is assumed.

jobname

identifies the job affected. The *jobname* must conform to the rules for VSPC jobnames (see the “Jobnames” entry). The entire *jobname* may be specified; alternatively, the *jobname* may be specified omitting the 6-character job entry code.

SMSG or DS(*dataset-identifier*)

when specified, requests that only the data set named be deleted.

SMSG specifies the system messages data set.

DS(*dataset-identifier*) specifies the data set for which *dataset-identifier* is the uniquely identifying portion of the SYSOUT data name (see the "LOAD Job Entry Command—to Retrieve Output" entry for instructions on determining the proper *dataset-identifier*).

When the DS operand is omitted, the entire job output identified by *jobname* is deleted.

When the SCRATCH command is executed, the batch system removes the specified job (or specified data set) from the VSPC online output class. If any of the job's output is in another output class, it is not removed. Thus, output previously sent to another output class or alternate destination via the ROUTE command cannot be deleted.

SEND Command

The SEND command transmits messages from the terminal to a specific destination.

SEND	' <i>message-text</i> '	Required first operand.
	CONSOLE OPERATOR <i>usernum</i>	Optional—choose one or omit. If omitted, OPERATOR is assumed.

'*message-text*'

represents the message to be transmitted; it can be made up of any combination of the following characters:

- the 26 alphabetic characters A through Z
- the 10 numeric characters 0 through 9
- blanks or spaces / + - * . , = () ' : ; ?

enclosed in single quotation marks. If the alphabetic characters are entered in lowercase, they are translated into capital letters, even if TRANSLATE OFF is in effect.

If the apostrophe or single quotation mark (') is contained within the message text, it must be specified as two adjacent characters (' '). In the message text, the adjacent characters are considered to be one.

'*message-text*' may be 1 through 64 characters long.

CONSOLE, OPERATOR, or *usernum*

when specified, identifies the message destination.

CONSOLE

specifies that the message is to be sent to the operating system console.

OPERATOR

specifies that the message is to be sent to the VSPC operator. If the destination operand is omitted, the message is sent to the VSPC operator.

usernum

specifies that the message is to be sent to the user so identified.

When *usernum* is specified, SEND command execution is unsuccessful if the receiving user is:

- in MESSAGE BLOCK mode
- not logged on

In either case, VSPC returns a message to the sender.

Note: If the message is not displayed at the receiving user's terminal before logoff, the message is purged.

SHARE Command

The SHARE command controls access to a file by users other than the owner.

SHARE	<i>filename</i> *	Optional first operand—choose one or omit. (Required if other operand is specified.)
	YES NO	Optional—choose one or omit. If omitted and the first operand is specified, YES is assumed.
	If all operands are omitted, the value for the current workspace is displayed.	

filename or *

specifies the file to which the SHARE command will apply.

filename

when specified, applies the SHARE characteristic to the named file. The *filename* must conform to the rules for VSPC filenames (see the "Filenames" and "Passwords" entries). The *filename* must not specify the special file CONTINUE.

*

when specified, associates the SHARE characteristic with the workspace. The SHARE characteristic does not take effect until the workspace is placed in a library.

YES or NO

specifies the SHARE characteristic to be applied.

YES

specifies that the file can be accessed by all VSPC users.

NO

specifies that the file can be accessed only by users with normal access to this library.

The SHARE NO command negates the effect of a previous SHARE YES command. Before any SHARE command is issued, SHARE NO is in effect.

After SHARE NO is specified for a private library file, only the owner may access the file. After SHARE NO is specified for a project library file, only users of that project library may access the file.

When a SHARE command is issued and both YES and NO are omitted, YES is assumed.

The SHARE command does not affect access to a public library file.

After a SHARE command is issued, it remains in effect until another SHARE command changes the SHARE characteristic.

For the effect of the SHARE command when specified with the PROTECT command, see Figure 16.

SPLIT Command

The SPLIT command separates a single line into two consecutive lines.

SPLIT	<i>line</i> *	Required first operand—choose one
	<i>number</i> 'string '	Required second operand—choose one

line

specifies the number of the line to be separated.

*

specifies that the current line is the line to be separated.

number

specifies the number of characters to be included in the first resulting line (the line-number and following blank are not included in the count).

If *number* is specified as 0 (zero), the first resulting line contains no characters.

If *number* is greater than the number of characters in *line*, the SPLIT command is rejected.

'string '

specifies a group of contiguous characters that are to begin the newly created line. If the string contains only alphabetic and numeric characters, but not all numerics, then the quotation marks may be omitted. However, if quotation marks are omitted, lowercase characters in the string will be automatically translated to uppercase for command entry. To get a string containing lowercase characters, specify TRANSLATE OFF and enclose the string in quotation marks.

If the specified character sequence occurs more than once in *line*, the first sequence is the point at which the new line begins.

General Execution Considerations

The line number assigned to the newly created line is calculated as follows:

- If the line-number following the original line is 20 or more higher, or if the original line is the last line in the workspace, the increment for the newly created line is 10.
- If the line number of the following line is 2 to 20 higher than that of the original line, the line number of the newly created line is mid-way between the two line numbers.

- If the line number of the following line is 1 higher than the original line, the newly created line is given a line number 1 higher than the original line and succeeding lines are renumbered with an increment of 1 as far as necessary to preserve line sequencing. In this case, the following message is returned

LINES RENUMBERED FROM *nn*

where *nn* is the first changed line number.

STATUS Job Control Command

The STATUS command queries the batch system about the current status of a previously submitted batch job.

STATUS	<i>jobname</i>	Optional operand. If omitted, status of all current jobs is queried.
---------------	----------------	--

jobname

identifies the job whose current status is being queried; *jobname* must conform to the rules for VSPC jobnames (see the "Jobnames" entry). The entire *jobname* may be specified; alternatively, the *jobname* may be specified omitting the 6-character job entry code.

General Execution Considerations

When the STATUS command is executed, the VSPC job entry facility passes the status request on to the batch processing system.

The system returns the information to VSPC, and one of the following messages is sent to the user:

Message	Meaning
<i>jobname</i> HELD	The specified job is in hold status.
<i>jobname</i> AWAITING EXECUTION	The specified job is in the input queue, but has not been initiated.
<i>jobname</i> EXECUTING	The specified job is currently being executed.
<i>jobname</i> COMPLETED	Execution of the specified job has been completed.
<i>jobname</i> NOT FOUND	The specified job is not in the system.
NO JOBS FOUND	There are no current jobs with VSPC-assigned jobnames for this user in the system.

STORE Command

The STORE command causes the source program in the workspace to be compiled, and the resulting object program to be placed in the specified file.

STORE	<i>filename</i>	Required operand.
	' <i>string</i> '	Optional

filename

specifies the name of the file in which the object program is to be placed; it must conform to the rules for VSPC filenames (see the "Filenames" and "Passwords" entries). CONTINUE must not be specified.

'*string*'

specifies a character string to be passed to the compiler. It may consist of 0 through 255 characters enclosed in single quotation marks. All characters are valid. However, if a single quotation mark (') is contained within the string, it must be specified as two adjacent single quotation marks (").

General Execution Considerations

The STORE command can be used to create a new object program file or to replace one that already exists.

The workspace must have a compiler *processor-name* as its attribute, either implicitly or explicitly.

When the STORE command is executed, the source program in the workspace is compiled by its associated compiler. If the compilation is successful, the resulting object program is placed in the specified file; it is identified by *filename*. After execution is completed, the source program in the workspace is unchanged.

If the STORE command creates a new object program file, that object program file can be placed in the user's private or project library, or any public library.

If the STORE command replaces a file that already exists, the existing file must be an object program file and only the file's owner can replace it.

During STORE command execution, an attention signal causes control to be passed to the compiler.

When compilation is successful, VSPC displays a header, the size of the object program, and the CPU time needed for compilation.

If a string is specified in the command, the use of that string is processor dependent.

SUBMIT Job Entry Command

The SUBMIT command enters a job or jobs for batch processing outside the VSPC environment.

SUBMIT	<i>filename-1</i>	Optional—specify or omit.
	<i>filename-n</i>	Optional—specify or omit. A total of 15 filenames may be specified. If all are omitted, workspace contents are assumed.

filename

when specified, is the source of the job stream. One through 15 filenames may be specified. Each must conform to the rules for VSPC filenames (see the “Filenames” and “Passwords” entries). Each identifies a file to which this user has access. If a file has PROTECT NOREAD in effect, it may be submitted only by its owner.

No *filename* may identify an object program or a CONTINUE file.

When all *filenames* are omitted, the workspace contents are submitted as a job stream.

General Execution Considerations

When the SUBMIT command is executed, the workspace or the specified *filenames* are submitted, in the same order as they are entered, as a job stream.

For the system in use, the job stream must be a complete and valid job or series of jobs.

Each record in the file is expanded to 80 characters, and its line number becomes its sequence number in the job stream. The placement of the sequence number within the 80 characters, or its omission from the record, may be specified by a NUMBER command (see “NUMBER Command”). If the user has not specified the placement of sequence numbers, the following default options are used:

- CLIST and DATA files, and source programs (except BASIC programs) omit sequence numbers from the records.
- BASIC source programs have line numbers of one through five digits, right justified on the fifth character position. (That is, the units position of the number is positioned at the fifth character in the record; leading zeros are included.) The sixth character position is a blank. BASIC source statements begin in the seventh character position.

If the number of digits in a line number exceeds the number of positions reserved for sequence numbering, SUBMIT command execution is terminated and an error message is issued. Also, the sequence number length plus the record length must not exceed 80 characters.

When more than one file is submitted in one job stream, each file has sequence numbers generated according to its explicit or implicit NUMBER specification. Thus it is possible to have mixed sequence number formats in one job. Lines exceeding 80 characters in length (including, if specified, the sequence number) cause SUBMIT command execution to terminate and an error message to be issued. Jobs already successfully submitted remain in the system; the job in error and all following jobs are not submitted.

The job stream for one SUBMIT command may contain 1 through 10 job statements. Every job statement in the stream is checked and completed by the SUBMIT command processor, including job statements in data streams (that is, job statements following DD DATA statements).

Under OS/VS1, the size of each job is limited to approximately 9,000 80-character records.

When a job is successfully submitted to the system, VSPC returns the following message:

JOB SUBMITTED AS *jobname*

jobname is in the form of a VSPC *jobname*—eight characters under OS/VS1; twelve characters under OS/VS2 (see the “Jobnames” entry).

If a job statement is detected following a DD DATA statement, VSPC returns the following message:

JOBCARD IN DATA HAS BEEN ASSIGNED JOBNAME *jobname*

jobname is in the form of an eight-character VSPC *jobname* under all host systems (see the “Jobnames” entry).

Each user-supplied job statement must reserve enough space to allow the eight-character *jobname* to be inserted. Any *jobname* in the user-supplied statement is replaced by the VSPC-assigned *jobname*. If an accounting identification is included in your user profile, that identification is inserted into the job statement.

If output is to be sent to the terminal, the proper JCL output class parameters must be included in SUBMIT command input. (The proper output class for any VSPC system is defined by the system administrator.) When the job output is retrieved for terminal display (by specifying LOAD OUTPUT), only this output class is accessed in retrieving it.

If attention is entered during SUBMIT command execution, then jobs already submitted remain in the job stream; jobs pending are removed. VSPC responds with the following message:

SUBMIT TERMINATED - *n* JOBS SUBMITTED

where *n* is a count of the number of already submitted jobs still in the job stream.

TABSET Command

The TABSET command communicates the terminal's session tab settings to VSPC.

TABSET	<i>tablist</i> OFF	Optional operand—choose one or omit.
	'x ' ' '	Optional operand—choose one or omit.
	EXPAND NOEXPAND	Optional operand—choose one or omit.
	If an operand is omitted, any previous setting is unchanged. If all operands are omitted, the current values are displayed.	

tablist

the list represents tab positions relative to the left margin of the line; it must consist of a list of 1 through 26 integers in ascending order. The left margin is usually position 1.

On a display terminal, in View Mode, the left margin is indented 6 columns and the assumed columns are as shown in the separator line. When using INPUT NOPROMPT in View Mode, position 1 is assumed to be the first character following each logical new line character.

When *tablist* is specified, the tab positions remain in effect until another TABSET command is successfully issued, or until the end of the session.

OFF

removes all current session tab settings.

'x '

specifies a character that VSPC will translate into the unprintable tab character. (On a 3270 terminal, the FIELD MARK key is also interpreted as a tab character.)

' '

(two adjacent quotation marks) specifies that no character will be translated into the tab character. (This operand can be used to remove a previously defined substitute tab character.)

EXPAND

specifies that input tab characters (or their substitute) are used to insert blanks into the data, as defined by the current tab settings.

NOEXPAND

specifies that input tab characters (or their substitutes) are placed in the workspace data as nonprintable characters. When the data is subsequently printed, the tab characters take effect in formatting the printed output.

At logon, TABSET OFF ' ' EXPAND is assumed.

General Execution Considerations

Physical and session tab settings must coincide, or output will be incorrectly formatted and, perhaps, even unreadable. That is, if the user changes the physical tab settings, but does not communicate these changes to VSPC or communicates them incorrectly, subsequent terminal operations will be incorrectly formatted.

The column tab settings are used to:

- Speed output to typewriter terminals (other than CPT-TWX) with hardware tabs
- Resolve output containing tab characters directed to an IBM 3270 screen or an IBM 3101 or CPT-TWX terminal and into an appropriate number of blanks
- Control expansion of real or substitute tab characters, on input, into strings of blanks if the EXPAND option is in effect
- Set the hardware tab settings on IBM SDLC 3767 and 3770 terminals

If a tab character is contained in output and either no tab columns are defined or the line position is beyond the last-set tab column, then the tab character is displayed as the logical tab character if one is set, or a blot character if not. The same is true of all tab characters contained in data displayed in the workspace display area in View Mode, or in the command area of the screen as a result of CHANGE with only a line number operand or a Program Function Key set to LAST.

For input, when TABSET NOEXPAND is in effect, a tab character (or its substitute) in a command string is treated as a separator in the command.

When TABSET OFF EXPAND is in effect, VSPC cannot interpret any tab keying encountered in input and the line is rejected; VSPC sends an error message to the user. (Some foreground processors take other actions. See the appropriate terminal user's guide.)

TAPE Command

For CPT-TWX terminals, the TAPE command specifies the paper tape reader as the input source.

TAPE		No operands allowed.
-------------	--	----------------------

After the TAPE command is executed, VSPC assumes the paper tape reader as the input source. Immediately before each input line, VSPC sends an X-ON character; after each input line no line feed is sent by VSPC. Whether or not terminal input is read from tape depends on the manual setup of the CPT-TWX terminal. If the setup is wrong and TAPE is specified, lines may be overprinted.

The KEY command specifies the keyboard as the input source, reversing the effect of the TAPE command. If no TAPE command has been specified, the keyboard is assumed to be the input source.

TEXT Command

The TEXT command presets actions to be taken by the CHANGE, FIND, and LOCATE commands.

TEXT	ON OFF	Optional operand—choose one or omit. If omitted, current value is displayed.
-------------	-------------------------	--

ON

specifies, if the affected commands do not explicitly specify TEXT or NOTEXT, that TEXT is assumed.

OFF

specifies, if the affected commands do not explicitly specify TEXT or NOTEXT, that NOTEXT is assumed.

General Execution Considerations

After the TEXT command is executed, any CHANGE, FIND, or LOCATE command issued without a TEXT/NOTEXT operand is treated as if the preset operand were implicitly specified.

Any such command explicitly specifying TEXT or NOTEXT uses the explicit specification rather than the implicit one.

Before a TEXT command is issued, NOTEXT is implicitly in effect.

TRANSLATE Command

The TRANSLATE command specifies:

- That a graphics substitution table, defined by the system administrator, is to be used
- Whether or not terminal input will have lowercase alphabetic letters translated into capital letters

TRANSLATE	<i>tablename</i> CAPS OFF APL	Optional operand—choose one or omit. If omitted, current value is displayed.
------------------	---	--

tablename

specifies that a table of graphic substitution characters has been defined by the system administrator, and that this *tablename* identifies it. (The character combinations CAPS and OFF are invalid as *tablenames*.)

CAPS

specifies that VSPC is to translate lowercase alphabetic characters entered from the terminal into capital alphabetic characters.

OFF

specifies that VSPC is to accept both lowercase and capital alphabetic characters exactly as they are entered.

APL

specifies that a start/stop terminal has an APL type element installed. If entered on a non-start/stop terminal, it is equivalent to TRANSLATE OFF.

One, and only one, operand must be specified.

General Execution Considerations

At logon, TRANSLATE CAPS is assumed, unless the logon command is entered on a start/stop terminal using an APL type element, in which case, APL is assumed.

The TRANSLATE specification remains in effect until another TRANSLATE command is successfully issued, or until the end of the terminal session.

Except for the character-string operands in quotation marks of the CHANGE, FIND, LOCATE and SPLIT commands, the TRANSLATE CAPS command has no effect on VSPC commands—whether entered from the keyboard or within a command list. When commands are entered in lowercase letters, VSPC automatically translates all translatable portions into capital letters.

VIEW Command

The VIEW command places the user in View Mode and/or displays another portion of the workspace without changing the current line pointer.

VIEW	<i>line</i> * – <i>line</i> + <i>line</i>	Optional operand—choose one or omit. If omitted, current line is displayed at top of screen.
-------------	--	--

line

specifies an absolute line number that is to be displayed at the top of the screen. The current line pointer is not changed.

*

specifies that the current line is to be displayed at the top of the screen.

– *line* and + *line*

specify a relative number of lines that the screen is to be moved.

– *line*

moves the display toward the beginning of the workspace

+ *line*

moves the display toward the end of the workspace.

General Execution Considerations

The VIEW command, if the user is in Command Mode, puts the user in View Mode and displays the specified line at the top of the screen. If all operands are omitted, the current line becomes the line at the top of the screen.

If the user is already in View Mode, the VIEW command without any operands specified places the current line at the top of the screen. (This is equivalent to issuing VIEW *).

(See also “Line Entry—View Mode.”)

VSPC (Logon) Command

The VSPC command connects the terminal with VSPC.

VSPC	ID= <i>usernum</i>	Required first operand.
	<i>x</i>	Required operand for S/S terminals. Ignored for others.

ID= *usernum x*
identifies the user to VSPC.

usernum

is assigned by the system administrator for each user of the subsystem.
It is a numeric identifier from one through seven digits long.

x

when specified, must be the upshift character of the "2" key. This operand is required for the following terminals: 3767 in start/stop mode, 1050, and 2741 and CPT-TWX-compatible terminals with APL keyboards. For other terminals, this operand is ignored.

General Execution Considerations

The VSPC command must be entered either entirely in capital letters or entirely in lowercase letters, as follows:

```
VSPC ID=12345  
vspc id=12345
```

No other combination of capital letters and lowercase letters is recognized as valid.

Notes:

1. Depending on the default your organization has chosen, you may or may not be able to abbreviate the VSPC command. Check with your system administrator.
2. For certain 3270 Display Systems that use the SDLC line discipline the TEST REQ key may be required instead of ENTER. If your IBM 3277 or 3278 is connected via a 3791, the procedure is also different. Consult *VSPC Terminals* or check with your VSPC administrator.

When the connection between the terminal and VSPC is made, either or both of the following two prompting messages may be received:

```
ENTER UP-SHIFT 2
```

When the ENTER UP-SHIFT 2 message is received, the *x* operand has been erroneously omitted, or incorrectly entered. The upshift character of the 2 key must be typed in response. If the response is incorrect, the prompt is repeated until logon is successful or the two-minute logon time limit has been reached.

```
ENTER PASSWORD
```

When the ENTER PASSWORD message is received, a password is required to complete a successful logon. The correct password must be typed in (see the "Passwords" entry for syntax rules). If a correct password is not supplied after four prompting messages, the logon is unsuccessful.

When logon is successful, VSPC sends some or all the following responses:

```
attribute hh:mm:ss mm/dd/yy usernum  
CONTINUE attribute
```

```
(or CONTINUE attribute PASSWORD)
```

```
logon-message
```

Note: The CONTINUE line is not always printed—see text.

attribute, in the first line, is the default workspace attribute for this user profile. Always displayed.

hh:mm:ss (hours:minutes:seconds) specifies the logon time. Always displayed.

mm/dd/yy (month/day/year) specifies the logon date. Always displayed. (This field may alternatively be displayed as *dd/mm/yy*—check with your VSPC administrators.)

usernum is this user's user number. Always displayed.

CONTINUE *attribute* or CONTINUE *attribute* PASSWORD appears only if there is a CONTINUE file in the user's library. The attribute is that of the CONTINUE file (it may be different from the default attribute). If a password is needed to obtain the CONTINUE file, the word PASSWORD is printed; otherwise it is omitted. This access password is the one that was set for the contents of the CONTINUE file.

logon-message contains a general information message. This line is displayed only if a logon message has been defined.

Note: If the workspace attribute or the CONTINUE file attribute is not presently known to VSPC, that part of the message contains asterisks (*). In this case, the workspace attribute is set to DATA.

APPENDIX A. TERMINAL FEATURES

Switch settings and actions taken for each valid VSPC terminal are shown in the following chart.

Action	IBM TERMINALS						CPT-TWX Mod 33/35*	
	3270	3101*	2741*	3767*		3770		1050*
Set switches for computer connection	Pull out OFF/PUSH Push the top (1) of the Power switch.	Push the right (1) of the Power switch. On high-pitched tone, press ALT, E	COM/LCL=COM ON/OFF=ON	Keylock=ON COMM/LOCAL=COMM AUTO/OFF=AUTO (SDLC only) EDIT/OFF=OFF (SDLC only) AUTO VIEW/OFF=as desired DOUBLE/SINGLE SPACE=as desired DATA/TALK=ask administrator DIAL/DISC=ask administrator SDLC/SS=ask administrator Primary/Secondary=as desired CALC/OFF=OFF TEST/OFF=OFF POWER/OFF=POWER PRESS SYS REQ (SDLC)	Keylock=ON POWER=ON HOLDPRINT=OFF EXTENT/ALARM=OFF PUNCH=OFF DISK=OFF UPDATE/MONITOR=OFF INTRP=OFF HALF SPEED=OFF or NORMAL SDLC/BSC=SDLC AUTO=ON (CODE/1 if 3771 or 3773) Press SYS REQ	MAIN-LINE=ON SYSTEM=ATTEND MASTER=ON PRINTER 1=SEND/REC PRINTER 2=HOME KEYBOARD=SEND PUNCH=NORMAL SYSTEM=PROGRAM EOB=AUTO SYSTEM=UP All others=OFF	Press ORIG button Model 35: Press K button On high-pitched tone, press CTRL. WRU	
Terminal ready to accept logon	SYSTEM AVAILABLE Cursor appears	Cursor advance	Keyboard unlocks	DATA SET READY, PROCEED lights on ONLINE (leased line)		PROCEED, KBD, LINE, and AUTO lights on	POWER, PROCEED lights on	Key mode: paper advance
Logon: VSPC ID=	usernum	usernum	usernum x	S/S usernum x	SDLC usernum	usernum	usernum x	usernum
End a line	ENTER	↵ (return)	RETURN	↵ (return)	↵ (return)	↵ (return)	RETURN ALTN CODING + EOT REQUEST	RETURN
Reset terminal	RESET	RESET	---	RESET	RESET	CODE and RESET READY Line number ON LINE and PROCEED light on	READY Line number Unlocks keyboard PROCEED light on	BRK-RLS Model 35: K READY Paper advance Line number Stops noise Bell
Terminal ready to accept input	READY Line number Cursor appears SYSTEM READY	READY cursor advance line number Bell	READY Line number Unlocks keyboard	READY Line number ON LINE and PROCEED lights on	READY Line number ON LINE and PROCEED lights on	← (backspace) INDEX, retype, ↵ (return) BUFFER RTN, CNCL	BACKSPACE, ATTENTION, retype, RETURN ATTENTION	Backarrow or underline, retype, RETURN CTRL/X
Correct current line	Position cursor, ERASE EOF	Underline, retype, RETURN	BKSP, ATTN, retype, RETURN	← (backspace), ATTN, retype, ↵ (return)	← (backspace). INDEX, retype, ↵ (return)	↵ (return) BUFFER RTN, CNCL	BACKSPACE, ATTENTION, retype, RETURN ATTENTION	Backarrow or underline, retype, RETURN CTRL/X
Delete current line(s)	ERASE INPUT	ALT/X	ATTN	ATTN	↵ (return) ATTN	↵ (return) ATTN	ATTN	ATTN
Send signal to system	PA1	BREAK	ATTN	ATTN	ATTN	ATTN	ATTENTION	BREAK
Escape from input	PA1	BREAK	ATTN	ATTN	ATTN	ATTN	ATTENTION	BREAK
End output	PA2	BREAK	ATTN	ATTN	CNCL	CNCL	ATTENTION	BREAK
Logoff	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
Turn off terminal	Push OFF/PUSH Push the bottom (0) of the Power switch.	Push the left (0) of the Power switch	ON/OFF=OFF	POWER=OFF Keylock=OFF	POWER=OFF Keylock=OFF	(unnecessary)	MAIN-LINE=OFF	CLR
Default line length	80	72	120	120	132	132	120	72
Invalid character prints as	" (double quote)	" (double quote)	(N backspace Z)	(N backspace Z)	(N backspace Z)	(N backspace Z)	(N backspace Z)	" (double quote)

*Terminals controlled by the Multiple Terminal Access (MTA) facility of the Network Control Program (NCP) may need special logon procedures. See your VSPC administrator.

APPENDIX B. HOW TO PRODUCE A LISTING OF ONLINE-MESSAGE EXPLANATIONS

The explanations of online messages are stored in a VSPC library and can be printed by running the VSPC service program ASUSPEXC. Online messages are messages received at the terminal from the interactive VSPC job in response to:

- General commands
- VSPC administrator commands
- VSPC operator commands
- VSPC monitor commands (these messages include the online log messages)

ASUSPEXC can be run at any time after VSPC is installed, whether the interactive VSPC job is running or not.

An example of OS/VS JCL for running ASUSPEXC to print the message explanations is:

```
//MESSAGES    JOB
//STEP        EXEC PGM=ASUSPEXC
//* STEPCAT FOR A VSAM USER CATALOG:
//STEPCAT     DD      DSN=VSAMCAT,DISP=SHR
//SYSPRINT    DD      SYSOUT=A
//SYSILDIR    DD      DSN=VSPCLIB.DIRECTRY,DISP=SHR
//SYSILIB1    DD      DSN=VSPCLIB.SOURCE,DISP=SHR
//SYSILIB2    DD      DSN=VSPCLIB.OBJECT,DISP=SHR
//SYSIN       DD      *
              MESSAGES 50
/*
```

In the example shown above, the AID prompts and message explanations are assumed stored in library 50. This should be changed to reflect the library number used in your organization. Similarly, the data set names for the VSPC library must be changed to correspond to those used in your organization.

The messages are listed in ascending sequence by message number. Variables in the text of a message are indicated by an expression beginning and ending with an asterisk. Following each message is an explanation. The first part of the explanation of each online log message identifies the type of log message: "LOG MESSAGE (URGENT, SESSION, or EXTERNAL)." For example:

```
ASU228      UNABLE TO SAVE 'CONTINUE'
            BECAUSE YOUR WORKSPACE WAS PROTECTED OR
            EMPTY, IT COULD NOT BE SAVED FOLLOWING AN OFF
            CONTINUE COMMAND OR FORCED ENDING.

ASU229I     *TIME* STOPPED BY CONSOLE OPERATOR
            LOG MESSAGE (URGENT). THE OPERATING SYSTEM
            OPERATOR HAS ISSUED A COMMAND TO STOP VSPC.
            VSPC WILL TERMINATE.
```

In a message itself, as received at the terminal, variables have specific values, in the appropriate form. For example, ***TIME*** is displayed in the form hh:mm:ss (hour, minute, second), and ***DATE*** is displayed in the form mm/dd/yy (month, day, year) or dd/mm/yy (day, month, year), according to the **DATE** startup control statement.

GLOSSARY

This glossary defines data processing terms used in this manual—including terms used only in VSPC as well as standard data processing terms. Terms not used in this manual are omitted.

American National Standard definitions are identified with asterisks (*).

IBM is grateful to the American National Standards Institute (ANSI) for permission to reprint its definitions from the *American National Standard Vocabulary for Information Processing* (Copyright © 1970 by American National Standards Institute, Incorporated), which was prepared by Subcommittee X3K5 on Terminology and Glossary of the American National Standards Committee X3.

access control: The alteration of normal VSPC file access conventions by use of the VSPC PROTECT and SHARE commands, and by the specification of passwords.

accounting records: Records in SMF format that contain information about various VSPC events for the purpose of accounting.

accounting unit: An installation-defined measure of VSPC system usage which is used for user accounting.

AID: See *VSPC AID*.

APL: A programming language. A problem solving programming language designed for use at terminals. It has special capabilities for handling arrays and for performing mathematical functions. VS APL is an IBM program product that can be used with VSPC.

ASCII:* American National Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

attribute: A descriptor of the kind of information contained in a user's workspace or files. A user profile is assigned a default attribute that applies to the user's workspace when he logs on to VSPC.

background: The portion of VSPC that provides batch processing capabilities.

BASIC: An algebra-like programming language used for problem solving by engineers, scientists, and others who might not be professional programmers. VS BASIC is an IBM program product that can be used with VSPC.

batch job: A serially processed job, as opposed to an interactive job.

batch processing: The processing of computer jobs serially, as opposed to interactive processing.

bulk input area: In View Mode, the area of the screen in which several workspace lines can be entered before interacting with the VSPC system to have the lines placed in the user's workspace.

byte: (1) * A sequence of adjacent binary digits operated upon as a unit and usually shorter than a computer word. (2) The representation of a character.

carrier return: The operation that prepares for the next character to be printed or displayed at the first position on the next line.

central processing unit: A computer unit that includes the circuits controlling the interpretation and execution of instructions. Abbreviated CPU.

character: A letter, digit, or other symbol that is used as part of the organization, control, or representation of data.

character string: * A string consisting solely of characters.

COBOL: Common Business-Oriented Language. A business data processing language.

command: An imperative statement used interactively, at a terminal, to tell VSPC to perform a specific action. *General commands* can be used by any VSPC user; *supervisory commands* can be used only by a VSPC administrator or the VSPC operator.

command list: A sequence of VSPC commands that can be saved for future execution as a group.

command list chaining: A method of defining VSPC command lists in such a way that execution of one command list automatically begins execution of another.

Command Mode: A VSPC processing mode that allows the user to enter general VSPC commands in the input area.

compilation: The process by which a foreground processor source program is translated into a machine language object program.

compile: To prepare a machine language object program from a source program written in another programming language.

compile time: The time during which a foreground processor source program is translated into a machine language object program.

compiler: * A program that compiles.

CONTINUE file: A special VSPC file, available to a VSPC user, in which the contents of the workspace are automatically placed when a forced ending occurs or when the user requests it during logoff.

CPU: central processing unit.

current line pointer: A line number within an editable workspace that is maintained by VSPC, that is set by commands which change the workspace, and that may be symbolically referenced in VSPC commands.

cursor: A movable spot of light on the cathode ray tube of a console or display unit that indicates where the next character will be entered.

data: * A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means.

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

default: The choice among exclusive alternatives made by VSPC when no explicit choice is specified by the user.

direct file: In VSPC, a file containing records of fixed length with line numbers beginning at one and incrementing by one. In such a file, specific records can be retrieved using the line number as a search key. A direct file can also be accessed sequentially.

directory: A set of records in the VSPC library, each of which identifies and describes a file in a user library. Each user library has its own directory.

editable data: One or more lines, each consisting of a line number followed by a character string; each line can be acted upon by the VSPC data editing commands. Editable data can consist of processor source programs, command lists, or simple data without any processing attribute.

external VSAM file: A VSAM file stored outside of the VSPC library that is available to VSPC users through language processor input/output statements.

file: A collection of related records treated as a unit; it is the unit of information stored in a user library.

file owner: The VSPC user who created a file. In a private library, the file owner and the library manager are the same user; in project and public libraries they may be either the same or different users. (See *library manager*.)

filename: A name that identifies a file. A filename consists of an optional library number, a required name field, and an optional password.

forced ending: A terminal session ending that is not under the control of the terminal user. A forced ending can occur when there is a detectable communication breakdown between the terminal and the computer system, when the system operator halts VSPC, or when the VSPC operator halts the user's session.

foreground: The portion of VSPC that provides interactive processing capabilities.

foreground processor: A computer program, such as a language processor, that runs interactively under the control of VSPC.

FORTTRAN: FORMula TRANslating system. A programming language used primarily to write computer programs in arithmetic formulas. VSPC FORTTRAN is an IBM program product that can be used with VSPC.

General Command Area: In View Mode, an area of the screen within which VSPC general commands may be entered.

general commands: VSPC commands that can be used by any VSPC user.

identifier: * A symbol whose purpose is to identify, indicate, or name a body of data.

integer: An unsigned natural number.

interactive processing: Computer processing in which each entry from a terminal elicits a response from the computer.

interpreter: A language processor that translates and executes each source program statement before it translates and executes the next one. For example, VS APL.

JCL: job control language.

job card: A job control language statement that begins and identifies a job within the job stream.

job control language: A programming language used to code job control statements. Abbreviated JCL.

job control statement: * A statement in a job that is used in identifying the job or describing its requirements to the operating system.

job entry: The submission of a batch computer job from a terminal.

job stream: On input, the sequence of job control statements and data submitted to the batch operating system; on output, the diagnostic messages and other output data produced when a batch job is executed.

jobname: An eight-character name that identifies a VSPC user's batch job. The first six characters are the user's preassigned VSPC job entry code; the last two characters are numeric and assigned by VSPC.

keyword: A VSPC command name or operand that identifies a specific action to be taken.

library: A collection of files. (See *VSPC library*, *private library*, *project library*, *public library*.)

library manager: A VSPC user whose user number is that of the associated user library. Such a user is authorized to remove a file from the specified library, even though he may not be the owner of the file.

library number: The user number that identifies a user's private, project, or public library.

light pen: A light-sensitive pen that can detect characters displayed on a 3270 cathode ray tube. By pressing the tip of the pen on control line fields, the user can request specific VSPC actions.

line number: A one-digit through five-digit number that begins a line of data. (See *editable data*.)

Local Command: In View Mode, a command that is entered over a line number in the workspace display area and that acts upon a limited number of lines.

logoff: A VSPC user's action of terminating a session with VSPC.

logon: A VSPC user's action of beginning a session with VSPC.

message: A combination of characters and symbols transmitted from one point to another on a telecommunication network.

message header: A six-character identifier for a VSPC message in the form ASU nnn , where nnn is a three-digit number. VSPC message headers are not normally printed; the user can, however, request that they be printed with each message displayed.

null line: A VSPC line that contains no characters. A null line is entered by pressing carrier return before any characters have been entered or after all characters have been erased.

object program: An executable computer program resulting from the compilation of a source program.

object program file: A VSPC file defined as containing a processor object program.

object time: The time during which a processor object program is executing.

operand: Information entered after a command name that modifies the action of the command and/or defines the data on which the command operates. (See *keyword*, *variable operand*.)

password: A string of one through eight characters that limits access to VSPC (logon password) or to a VSPC file (file password).

PF key: Program function key.

private library: A VSPC library containing files that are normally available only to the user with that library number (user number); if this user so specifies, however, such files may be made available on a read-only basis to other VSPC users.

processor: A computer program that runs either interactively under the control of VSPC (*foreground processor*) or as a separate batch job that communicates with VSPC (*auxiliary processor*).

profile: See *user profile*.

program function (PF) key: A key on the keyboard of a display device that passes a signal to a program to call for a particular program operation.

program product: A licensed program that performs a function for the user and usually interacts with and relies upon system control programming or some other IBM-provided control program. For example, VSPC is an IBM program product that relies on OS/VS1, OS/VS2, or DOS/VS.

programming language: A language used to write computer programs, such as APL, BASIC, or FORTRAN.

project library: A VSPC library containing files that are normally available only to a subset of VSPC users whose profiles specify that they may have access to the library. Files in this library may be modified only by their owners. (Ownership of files may be transferred among project library users.) Files may be removed only by the owner or the library manager.

prompt: A request from VSPC for certain information, such as a password.

public library: A VSPC library containing files that are normally available to all VSPC users. Files in this library may be modified only by their owners. Files may be removed only by the owner or the library manager.

S/S mode: Start-stop mode.

SDLC mode: Synchronous data link control mode.

sequence number: A number that identifies the position of a batch job record in the sequence of records for the job.

sequential file: A file whose records are organized on the basis of their successive physical positions, such as on magnetic tape. Contrast with *direct file*.

session: The activity that occurs at a terminal between logon and logoff.

source program: A sequence of computer program statements written in a programming language and requiring compilation or interpretation to be executable by the computer.

source time: The time during which a language processor source program is created.

special character: * A graphic character that is neither a letter, nor a digit, nor a space character.

start-stop mode: Asynchronous transmission between a terminal and the computer in which each character (or group of code) is preceded by a start signal and followed by a stop signal. Abbreviated S/S mode.

string: * A linear sequence of entities such as characters or physical elements.

synchronous data link control mode: A type of line control that uses a high-speed synchronous data communication technique. Abbreviated SDLC mode.

system administrator: Any person responsible for the design and/or available facilities of the computer system under which VSPC operates. The system administrator assigns system-dependent operand values, such as the CLASS and DEST values in a ROUTE command.

terminal: A device, usually equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel.

undefined file: A VSPC processor output file that has records in a format unrecognizable by VSPC; such files cannot be edited using VSPC.

unit-number: An identifying number given to a VSPC file through the ALLOCATE command; the unit-number corresponds to the dataset-number used in FORTRAN input/output statements.

user number: The number that identifies a user to VSPC.

user profile: The definition of the characteristics associated with a specific end user, including: the user number, the optional password, whether or not this user is a project or public library manager, maximum storage allocation for this user's library and workspace, the default workspace contents, whether or not this user is able to use job entry commands, and whether or not this user is privileged to use administrative commands.

variable operand: An operand that modifies the action of a VSPC command through a user-chosen value.

View Mode: A VSPC display terminal processing mode in which the user can modify a displayed workspace using the local facilities of the terminal, and using local and general commands, and can then copy changes back into the workspace.

VS Personal Computing (VSPC): An IBM program product that allows interactive or batch processing from a terminal. Abbreviated VSPC.

VSAM: Virtual storage access method. The set of programs the operating system uses to transfer data between VSPC and direct-access storage.

VSPC: VS Personal Computing.

VSPC administrator: A VSPC user authorized by an indicator in his user profile to use the supervisory commands for enrolling new VSPC users and processing their user profiles. He also has special authority to use the VSPC service program. He is either a *chief administrator* or an *account administrator*.

VSPC AID: A program contained in VSPC that, on request, prompts a user to specify a VSPC command correctly, or provides an explanation of a VSPC message.

VSPC library: A set of three VSAM data sets that contain *profiles* and *user libraries* and their *directories*.

VTAM: Virtual telecommunications access method. The set of programs the operating system uses to transfer data between terminals and VSPC.

workspace: The storage in which a VSPC user does his terminal work (such as entering data or source program statements and compiling and executing programs).

workspace display area: An area of a display screen in View Mode in which part of the user's workspace is displayed and can be changed by use of the local facilities of the terminal.

INDEX

Symbols

. (see period)
() (see parentheses)
+ (see plus sign)
\$ (see currency symbol)
* (see asterisk)
- (see hyphen)
- (see minus sign)
/ (see slash)
/"nn prelogon entry for 2741 33
, (see comma)
? (question mark) in AID
 commands 64,99,107
: (see colon)
@ (at sign) (see upshift 2 character)
' (see apostrophe)
' (see quotation mark)
' (invalid output character) 38
= (equal sign) valid in message
 text 169

A

A through Z in name field 23,122
abbreviations
 of commands 20,105
 of keyword operands 20,104,105
 of keywords 20,104
about this book 3
access control
 (see also password)
 definition 187
 extending access 79,170
 general description 25
 restricting access 78,153
 summary of command actions 131
access control commands
 file status information 131
 general description 25
 LIST command displays 135
 LOAD command execution
 and 136
 MERGE command execution
 and 141
 PASSWORD command 53,149
 PROTECT command 78,153
 SHARE command 79,170
access control features
 file protection 25
 file sharing 26
 passwords 25
access control summary 131
access rules
 private libraries 23,130
 project libraries 23,130
 public libraries 23,131
accesses, terminal, obtained by
 QUERY 160
accounting information, QUERY
 command obtains 160
accounting records, definition 187
accounting unit, definition 187
ACQUIRE command
 description 107
 examples 80
 format 107
 usage of 80
adding characters to a line 69
adding files to a workspace 141
AID (see VSPC AID)
AID commands, general
 description 19
ALLOCATE command
 description 113
 examples 86
 format 110
 FREE command and 86,125
 overriding 86,125
 usage of 86
alphabetic characters
 in name field 23,122
 in passwords 150
 valid in message text 169
 when accepted as entered 178
 when translated to capitals 58,178
alphabetic data, translation
 of 58,59,178
alphabetic first character in name
 field 23,122
alphanumeric characters in name
 field 23,122
alphanumeric order, of QUERY
 LIBRARY command
 display 158
alternate job output
 destinations 95,165
American National Standard Code for
 Information Interchange
 (ASCII) 187
American National Standard glossary
 definition 187
American National Standards
 Institute,
 acknowledgment 187
ANSI (American National Standards
 Institute),
 acknowledgment 187
APL, definition 187
APL files
 file status information 160
 QUERY LIBRARY command
 and 159
apostrophe (')
 (see also quotation mark)
 used in operands 20,106
 valid in message text 62,169
appendix of terminal features 183
assigning FORTRAN files 86,110
assigning sequence numbers 76,147
assumed line lengths 54,134
asterisk (*)
 ALLOCATE command
 and 86,110
 CHANGE command and 113
 COPY command and 115
 DELETE command and 116
 EXTRACT command and 118
 FIND command and 124
 HARDCOPY command and 126
 identifies ANSI glossary
 definition 187
 in formats 106
 in LIST command execution 135
 in logon messages 181
 INPUT command and 126
 JOIN command and 129
 LIST command and 135
 LOCATE command and 140
 MERGE command and 141
 MOVE command and 144
 NUMBER command and 147
 PROTECT command and 153
 RENUMBER command and 164
 SHARE command and 170
 specifies terminal 86,110
 SPLIT command and 171
 valid in message text 169
 VIEW command and 179
ASUnnn as message header 63
at sign (@) (see upshift 2 character)
attention key, using
 canceling output 36
 ending automatic line
 numbering 45
 escaping from input 36
 signaling system 36
attention signal
 ends automatic line numbering 45
 interrupts job submission 93,175
 STORE command execution
 and 173
 terminates batch job
 submission 93,175
 WAIT message mode and 63,143
attribute
 changing 63,117
 CLEAR command does not
 change 73,115
 definition 187
 description 18
 example of usage 30
 for files 23
 LOAD command changes 137
 logon message displays 34,178
 MERGE command does not
 change 142
 QUERY PROFILE command
 displays 60,161
 QUERY WORKSPACE command
 displays 162
 RUN command restriction 167
automatic line numbering
 description 43,126
 ending 43,126

B

background, definition 15,187
backspace
 as editable data character 132
 in line correction 35
BACKSPACE command
 description 111
 examples 57
 format 111
 usage of 57
BASIC, definition 187
BASIC files
 file status information 161
 QUERY LIBRARY command
 and 159
 retrieval 136
BASIC source programs
 batch job numbering 174
 job entry considerations 91
batch job, definition 187
batch processing
 availability of 90
 canceling 97,112
 definition 187
 erasing output 97,168
 general description 90
 job submission 174
 programming
 considerations 90,174
 querying status 93,172
 remote output 96,165
 terminal output 94,136
batch processing commands (*see* job
 entry commands)
blank
 as separator 21,103
 in line entry 132
 valid in message text 169
 when required in data line 132
blocking message receipt 63,143
bulk input area 47,49,187
byte, definition 187

C

CANCEL job entry command
 description 112
 examples 97
 format 112
 OS/VS1 considerations 97,112
 OS/VS2 considerations 97,112
 SUBMIT command and 112
 usage of 97
canceling a batch job 97,112
capital letters
 in examples 30
 in passwords 150
 translation to
 in commands 30
 in data 58
 TRANSLATE command
 and 58,178
CAPS invalid as tablename 178
carrier return (cr)
 definition 187
 ends automatic line
 numbering 45,126

ends line entry 35,132
implied in examples 30
line correction and 35
line deletion and 43
removes password 149
central processing unit (CPU)
 definition 187
 QUERY command displays
 statistics 160
CHANGE command
 adding characters 69
 deleting characters 69
 description 113
 examples 67-69
 format 113
 PROTECT command
 restriction 154
 TRANSLATE command
 exception 178
 usage of 67-69
changing a range of lines 68,113
changing a single line 68,113
changing attribute explicitly 60,117
changing current line pointer
 examples 31-32
 local command 50
changing line increment
 INPUT command 43,126
 MERGE command 74,141
 MOVE command 65,144
 RENUMBER command 70,164
changing line numbering 70,164
changing lines in entire
 workspace 67,113
changing logon password 53,149
changing profile, and QUERY
 PROFILE 161
character, definition 187
character string
 adding 69
 CHANGE command use 67,113
 definition 20,187
 deleting 69,113
 differentiation of 68
 FIND command use 69,124
 passwords 150
 PFKEY command use 38
 quotation mark in 69
 replacing 68,113
 SEND command use 62,169
 TRANSLATE command
 exception 58,178
character translation
 in commands 33
 in passwords 25,150
 TRANSLATE command
 and 58,178
characters
 added by PUNCH ON 156
 invalid output 36,38
 number displayed by QUERY
 WORKSPACE 162
 valid for
 filenames 122
 message text 169
 passwords 150
checking VSPC usage

accounting information 160
 profile 59,161
 tabs 59,160
 translation convention 160
 workspace statistics 162
class, output, ROUTE command
 specifies 96,165
CLEAR command
 description 115
 example 73
 format 115
 sequence numbering and 148
 usage of 73
clearing a workspace 73,108
CLIST attribute
 command list processing and 87
 definition 18
 ENTER command and 117
 QUERY LIBRARY command
 and 159
CLIST files (*see* command list files)
CLIST processing (*see* command list
 processing)
COBOL
 batch processing and 91,92
 command lists and 88
 definition 187
collating QUERY sequence, EBCDIC,
 in QUERY LIBRARY
 command 159
colon (:)
 denotes range of
 numbers 20,65,106
 in formats 106
 used in operands 20,65
 valid in message-text 169
column tab settings
 TABSET command 177
combining files (*see* MERGE
 command)
combining PROTECT command
 operands 79,153
comma (,)
 as a separator 20,103
 valid in message text 169
command
 abbreviations 20,102,104
 continuation 104
 correcting 35
 definition 187
 deleting 35
 description 16,19-21
 format notation of 105,106
 local, definition 188
 operand usage 20
 separators 21,105
 syntax of 103
command area, general 47,188
command functions, list of 102
command list
 as data 22
 chaining of 90
 chaining of, definition 187
 definition 187
 ENTER command and 83,117
 examples 87-90
 file status information 156

RUN command executes 89,166
processing of 87-90
command list files
batch job numbering 174
creation 87
execution 89
retrieval 136
command list processing
chaining 90
creation 87
creation example 88
description 87-90
execution 89,166
execution example 89
nesting 90
password and 150
VPSC AID during 109
command mode
definition 187
description 41
line entry 132
command-name
abbreviations 20,104
in formats 105
syntax 103
comments 102
compilation
definition 187
RUN command performs 89,166
STORE command
performs 84,173
compile, definition 187
compile time
definition 187
description 83
compiler
definition 187
description 83
messages 84
RUN command invokes 89,166
STORE command invokes 84,173
compiling and saving source
programs 84
complete job output,
retrieving 93,94,136
computer contact before logon
leased line 33
switched line 33
concepts of VSPC 15-26
connecting to VSPC 33
console operator, message
recipient 169
contact procedure
leased line 33
switched line 33
content attribute (*see* attribute)
continuation of commands 104
CONTINUE file
definition 187
description 123
in logon message 180
NAME command restriction 145
OFF command and 61,149
PROTECT command invalid 153
SAVE command restriction 167
STORE command restriction 173
SUBMIT command restriction 174

use of 60,61,123
continuous message receipt 63,143
control area 46,47
control line, description 41
controlled project library
defined 24,130
controlled public library
defined 24,131
conversational processing,
definition 15
COPY command
description 115
examples 66
format 115
usage of 66
correcting data 35,68,113
CPT/TWX terminals
default line length 134
description 30
KEY command 129
keyboard assumed
input source 129
PUNCH command 155
restriction on input 35
summary of features 183
TAPE command 177
CPU (*see* central processing unit)
CPU time, obtained by QUERY
command 160
cr (*see* carrier return)
creating files
description of 75
FILE command 119
creation
of files 75,119
of logon password 149
of object programs 85,173
of passwords 25,149
currency symbol (\$) in name field 123
current line
correcting 36
deleting 35
current line pointer
changing 31,50
definition 187
description 31
EXTRACT command and 73
FIND command and 69
LOCATE command and 31
cursor
definition 187
positioning 52
requesting printing with 126

D

data
alphabetic, translation
of 58,59,178
changing in workspace 67,113
definition 188
description of 16,21,22
examples 22
forms of 21
unexpected endings and 22
usage 22
when automatically
saved 22,60,149
when erased during
session 22,64,116
when saved at end of
session 22,60,72,149
DATA attribute
definition 18
direct files must have 121
ENTER command and 117
FILE command use of 119
QUERY LIBRARY command
and 158
usage in examples 31,36
data editing commands
general description 19
usage 41-45,64-73
data editing, usage of 41-45,64-73
data entry
deleting lines 43
general description 21,22
line-numbered 42
usage of 41-45,64-67
data files
batch job numbering 77,147,174
CONTINUE file disallowed 123
file status information 81,156-157
QUERY command and 81,156-157
retrieval 74,136
storage 72,167
data line
CHANGE command alters 69,113
changing current 35
DELETE command erases 64,116
deleting current 35
FIND command displays 69,124
data set, definition 188
data stream, job statements in 175
dataset-identifier
LOAD command and 94,136
ROUTE command and 95,165
SCRATCH command and 98,168
date field
in file status information 81,156
in logoff message 61
in logon message 34,181
date file last read, file status
information 81,156
date file last written, file status
information 81,156
DD DATA statements, job statements
following 175
dd/mm/yy (*see* date field)
default
batch job numbering 174
definition 188
increment in renumbering 164
line lengths 134
output class in ROUTE
command 95,165
SHARE command 170
DELETE command
description 116
examples 64
format 116
usage of 64
deleting characters from a line 64,113
deleting files 81,156
deleting job output 97,168

- deleting lines
 - during data entry 35,64
 - from workspace 64,116
 - local command 51
 - logon password 33,140
 - tab settings 176
 - workspace contents
 - CLEAR command 73,115
 - OFF command 149
- destination line
 - MERGE command and 141
 - MOVE command and 144
- dial environment
 - logon and 33
 - OFF command and 149
- dictionary
 - format presentation in 105,106
 - general user's 101-181
 - organization of entries 101
- differentiating character strings 68
- direct file
 - definition 188
 - description 121
 - FILE command and 119
 - file status information 156
 - general description 24
 - MOVE command warning 145
 - NUMBER command and 147
 - QUERY command and 157,158
 - record size displayed 156
 - renumbering restriction 165
 - SAVE command warning 167
 - sequence numbering for 147
- direct undefined file
 - description 121
 - FILE command and 119
 - file status information 156
 - QUERY command and 157,158
- directory, definition 188
- display
 - FIND command conventions 124
 - line length specification 134
 - line numbers 126
 - STORE command messages 173
- display area, workspace 47,48
- dollar sign (\$) in name field 122

E

- EBCDIC collating sequence, QUERY command and 159
- editable data
 - changing line numbering 70,164
 - changing lines of 67,113
 - definition 64,188
 - displayed by LIST command 72,135
 - ENTER command and 117
 - entry of 35,64-67,132
 - erasing 64,116
 - moving 65,144
 - searching workspace for 69,124
 - size obtained by
 - QUERY PROFILE command 161
 - QUERY WORKSPACE command 162

- editing
 - command mode 41
 - view mode 46
- editing line numbers 126
- editing range of lines 68,113
- editing single line 68,113
- ending automatic line numbering 43,126
- ending terminal session 61,149
- ENTER command
 - CLEAR command and 115
 - command lists and 88
 - description 117
 - examples 60
 - foreground processors and 83
 - format 117
 - list of processor-names 117
 - usage 60,88
- ENTER INPUT (3270 terminal) 38
- ENTER key of 3270 requests printing 126
- entry
 - command mode 41
 - view mode 46
- entry, 2741 optional logon 34
- equal sign (=) valid in message text 169
- erasing data 64,116
- erasing entire workspace 73,115
- erasing files 82,156
- erasing job output
 - CANCEL command 97,112
 - SCRATCH command 98,168
- erasing partial workspace 73
- erasing tab settings 176
- erasing workspace lines 64,116
- erasure of messages at logoff 149
- examples
 - ? command 64,98,99
 - ALLOCATE command 86
 - CANCEL job entry command 97
 - CHANGE command 67
 - CLEAR command 73
 - DELETE command 64
 - ENTER command 60
 - FEATURE command 118.1
 - FILE command 75
 - FIND command 69
 - format of 30
 - FREE command 86
 - HARDCOPY command 40
 - INPUT command 43
 - LINESIZE command 54
 - LIST command 72
 - LOAD job entry command 74,94
 - LOCATE command 31
 - MERGE command 74
 - MESSAGE command 62
 - MOVE command 65
 - NAME command 71
 - NUMBER command 76
 - OFF command 61
 - PASSWORD command 53
 - PFKEY command 38
 - PROTECT command 78
 - PURGE command 82
 - QUERY command 59,81

- RENUMBER command 70
- ROUTE job entry command 95
- RUN command 85,89,90
- SAVE command 72
- SCRATCH job entry command 98
- SEND command 62
- SHARE command 79
- special characters in formats 106
- STATUS job entry command 93
- STORE command 84
- SUBMIT job entry command 92
- TABSET command 55
- terminals used for 30
- TRANSLATE command 58
- VSPC (logon) command 33
- executing a program 85
- execution time (*see* object time)
- explanatory information (*see* comments)
- extending file access 79,170
- external VSAM file
 - definition 188
 - description 25,122
- EXTRACT command
 - description 118
 - examples 73
 - format 118
 - usage of 73

F

- facilities of VSPC 15-26
- FEATURE command
 - description 118.1
 - format 118.1
- file
 - access control summary 131
 - creation
 - by FILE command 76,119
 - by SAVE command 72,167
 - by STORE command 84,173
 - definition of 188
 - deletion 81,156
 - merging 74,141
 - restricting access
 - with file passwords 78,123,150
 - with PROTECT command 78,153
 - sharing 79,170
 - transferring 80,163
- file access control features
 - general description 25,26
 - passwords 122,150
 - PROTECT command 78,153
 - SHARE command 79,170
 - summary of 131
- file attribute, definition 23
- FILE command
 - CONTINUE disallowed 119
 - description 119
 - examples 76
 - file status information 156
 - format 119
 - usage of 76
 - when password required 149

- file creation
 - FILE command 75,119
 - file status information 156
 - SAVE command 72,167
 - STORE command 84,173
 - file length, object file status information 156
 - file modification
 - FILE command and 76,119
 - PROTECT NOWRITE prevents 78,154
 - file organization
 - detailed description 121
 - general description 24
 - file owner
 - definition 23,188
 - description 130,156
 - file status information 23
 - of project library 130
 - of public library 131
 - PURGE command specification and 156
 - QUERY command and 156,158
 - RELEASE command and 163
 - STORE command and 173
 - file password
 - detailed description 150
 - filename usage of 23,123
 - general description 16,25
 - PROTECT WRITE can omit 153
 - PURGE command can omit 156
 - file protection
 - passwords and 16,25,123,150
 - PROTECT command and 26,153
 - summary of access control 131
 - file replacement, SAVE command and 167
 - file retrieval
 - description 136
 - file status information 156
 - file sharing, description 26,170
 - file size
 - FILE command specification 119
 - limit in file status information 156
 - file status information display 156
 - file transfer 163
 - file type, in status information 156
 - file types
 - detailed description 121
 - general description 24
 - file update
 - FILE command 75,119
 - file status information 156
 - SAVE command and 72,167
 - filename
 - ALLOCATE command operand 110
 - CLEAR command and 115
 - CONTINUE special file 123
 - definition 188
 - detailed description 122
 - FILE command operand 119
 - general description 23
 - identifies library files 16
 - LOAD command operand 136
 - MERGE command operand 141
 - NAME command operand 145
 - names workspace 145
 - NUMBER command operand 147
 - passwords for 150
 - PROTECT command operand 153
 - PURGE command operand 156
 - QUERY FILE command operand 156
 - QUERY WORKSPACE command displays 162
 - RUN command operand 166
 - SAVE command operand 167
 - SHARE command operand 170
 - STORE command operand 173
 - filenames, fifteen allowed in SUBMIT command 92,174
 - files
 - assigning FORTRAN 86,110
 - creating and
 - updating 72,75,119,167
 - deleting 82,156
 - insertion of 74,141
 - interleaving of 75,141
 - library storage 130
 - obtaining 74,136
 - preventing modification 78,153
 - printing information about 81,156
 - releasing FORTRAN 86,125
 - removing restrictions on 79,153
 - restricting access to 78,153
 - transferring 80
 - FIND command
 - description 124
 - examples 69
 - format 124
 - PROTECT command restriction 154
 - TRANSLATE command exception 178-179
 - usage of 69
 - FLINK (FORTRAN link processor) 117
 - first character in name alphabetic 122
 - forced ending
 - CONTINUE file and 61,123
 - definition 188
 - foreground, definition 15,188
 - foreground processor
 - definition 15,188
 - description 82
 - ENTER command and 83,117
 - incorrect passwords and 151
 - publications references 3
 - TRANSLATE command and 58
 - usage of 82-87
 - foreground processor files
 - ALLOCATE command and 86
 - file status information 156
 - LOAD command and 136
 - QUERY command and 156,159
 - format
 - of commands 19-21,105
 - of examples 30
 - format notation, description 105,106
 - format presentation, illustration 105
 - FORTRAN, definition 188
 - FORTRAN link processor 117
- F**
- FORTTRAN files
 - ALLOCATE command and 86,110
 - batch job numbering of 174
 - command list processing for 87
 - file status information 156
 - FREE command and 86,125
 - QUERY FILE command and 156
 - retrieval 136
 - FREE command
 - description 125
 - examples 86
 - format 125
 - usage 86
 - full screen editing (*see view mode*)
 - functions of commands, list 102
- G**
- general command area 47,188
 - general commands, definition 188
 - general messages, and MESSAGE mode 143
 - glossary 187-190
 - graphic characters, all valid 132
 - graphics substitution table specification 178-179
- H**
- handling workspace data 41-45,64-73
 - HARDCOPY command
 - description 126
 - examples 40
 - format 126
 - usage of 40
 - HCPY control field, requests printing 126
 - header area, description 41,47
 - header for messages described 63
 - hh:mm:ss (*see hours:minutes:seconds*)
 - highest line number displayed by QUERY WORKSPACE 162
 - hours:minutes:seconds (hh:mm:ss)
 - logoff message 61
 - logon message 34,180
 - QUERY command message 161
 - hyphen (-) in formats 106
- I**
- IBM terminals
 - features chart 183
 - 1050 description 30
 - 2741 description 29
 - 3270 description 27
 - 3767 description 27
 - 3767/2741 used in examples 31
 - 3770 description 29
 - identifier
 - data line 21,22
 - definition 188
 - message 63
 - incorrect password prompting 151
 - increment, control of
 - INPUT command 43,126
 - MERGE command 141
 - MOVE command 143

RENUMBER command 164
increment, default 126,141,143,164
information as data 21
input acceptance signals 35
input area, bulk 47,49,187
input area, description 41
INPUT command
 attention signal ends
 execution 43,127
 carrier return ends
 execution 43,127
 description 126
 examples 43
 format 126
 replacing lines using 43
input lines, procedure for 35
input tab formatting 55,176
input, TAPE command for 177
inserting files into workspace 74,141
integer, definition 188
interactive processing
 definition 188
 description 15
interleaving files (*see* MERGE
 command)
interpreter
 definition 188
 description 82
interpreter files, LOAD command
 invalid 136
interpreter workspace size, QUERY
 PROFILE and 161
interrupting job submission 93,175

J

JCL (*see* job control language)
job control language (JCL)
 definition 188
 examples 92,93
 in batch processing 92
 LOAD command and 136
 parameters for terminal
 output 175
job control statement, definition 188
job entry, definition 188
job entry code
 batch processing and 89,128
 jobname and 128
 profile defines 16
 QUERY PROFILE displays 155
 ROUTE command and 158
 SCRATCH command and 168
 STATUS command and 165
 SUBMIT command and 166
 used in examples 29
job entry commands 17
job entry facilities
 availability of 90
 canceling jobs 97,112
 erasing output 98,168
 general description 15,90
 job submission 92,174
 programming
 considerations 91,92,174
 querying status 93,172
 remote output 96,165
 terminal output 94,136

job identifier, in jobname 128
job output
 erasing 98,168
 remote destination 96,165
 terminal as destination 94,136
job statement (*see* job control
 statement)
job status messages 93,172
job stream
 definition 188
 source specification 174
jobname
 CANCEL job entry commands
 and 112
 definition 188
 description 128
 LOAD command and 136
 ROUTE command
 specification 165
 SCRATCH command and 168
 STATUS command and 172
 SUBMIT command messages 175
 VSPC defines 175
JOIN command
 description 129
 examples 67
 format 129
 usage of 67

K

KEY command
 description and format 129
 TAPE command and 177
keyboard, assumed CPT-TWX input
 source 129
keyboard unlocked to receive
 input 35
keyword
 abbreviations 20,104
 definition 188
 operands
 definition 20,105
 in formats 105
 overriding 103

L

language processors
 description of use 82-87
 incorrect passwords and 151
last line displayed by QUERY
 WORKSPACE 162
leased line
 OFF command and 149
 procedure for using 33
left margin sequence numbers
 file status information 157
 NUMBER command and 147
length of workspace displayed by
 QUERY WORKSPACE 162
libnum in filenames
 description 23,122
 library identification 130
 QUERY FILE command and 158
libraries
 characteristics removed by CLEAR
 command 73,115

detailed description 130
naming conventions for 71,145
obtaining directories 81,158
placing command lists in 87
placing source programs in 84
profile, definition of 18
QUERY PROFILE and size 161
types used in examples 31
usage obtained by QUERY
 command 160
 usage of 74-82
library access
 passwords for 150
 QUERY command and 158
library, definition 188
library files, NAME command
 warning on 145
library management commands 19
library manager
 definition 23,130,188
 of project library 130
 of public library 131
 PURGE command
 specification 156
library number
 definition 23,188
 description 23,122
 syntax 122
 when omitted 23,122
library profile entry, description 18
library type, QUERY PROFILE
 displays 150
light pen
 definition 188
 requesting printing with 126
limits (*see* maximum)
line addition, local command 50
line deletion
 during data entry 35,37,43
 in workspace 43,116
 local command 51
line entry
 command mode 132
 detailed description 132
 general description 21,22
 INPUT command and 43,126
 order of 42
 procedure 35,132
 view mode 132
line feed, TAPE command
 suppresses 177
line increment
 default
 INPUT command 43,126
 MERGE command 141
 MOVE command 144
 RENUMBER command 164
 specification
 INPUT command 43,126
 MERGE command 141
 MOVE command 144
 RENUMBER command 164
line length
 batch job maximum 175
 restriction on specification 134
 specification 54,134
 too long 48

- LINE NOT FOUND 141
 - line numbered data entry 42,132
 - line numbering
 - accepting input 35
 - automatic 43,126
 - begins data line 21,132
 - by programmer 44
 - changing increment 70,164
 - changing workspace 70,164
 - ending automatic 45,126
 - RENUMBER specification 70,164
 - line numbers
 - definition 188
 - detailed description 132
 - direct file rules for 121
 - display of 135
 - excess digits cause error 174
 - general description 21,22
 - LOAD command and 137
 - number of displayed by QUERY WORKSPACE 162
 - overlap invalid 164
 - sequence numbers and 174
 - specification 147,164
 - suppressing display 44,126
 - use of described 42-45
 - line overlap
 - INPUT command and 43,126
 - MERGE command and 141
 - MOVE command and 144
 - RENUMBER command, invalid for 164
 - line pointer, current (*see* current line pointer)
 - line range
 - CHANGE command 113
 - COPY command 66
 - DELETE command 116
 - FIND command 124
 - MOVE command 144
 - View Mode 48
 - line reformatting 67
 - line renumbering
 - INPUT command and 43
 - MOVE command and 65,144
 - NUMBER command 147
 - RENUMBER command 70,164
 - line repetition, local command 51
 - LINESIZE command
 - description 134
 - examples 54
 - format 134
 - restriction 134
 - usage 54
 - LIST command
 - description 135
 - examples 72
 - format 135
 - PROTECT command
 - restriction 153
 - usage of 72
 - list sequence numbers 147
 - LOAD command
 - (*see also* LOAD job entry command)
 - CONTINUE file retrieval 123
 - description 136
 - examples 74
 - file status information 156
 - format 136
 - PROTECT command
 - restriction 153
 - usage of 74
 - when password required 150
 - LOAD job entry command
 - (*see also* LOAD command)
 - description 136
 - examples 94,95
 - format 136
 - OS/V5 considerations 94,138
 - sequence numbering and 148
 - usage of 94,95
 - local command, definition 188
 - Local commands
 - adding lines 50
 - changing current line pointer 50
 - deleting lines 51
 - description 138
 - examples 50-51
 - format 138
 - order of processing 51
 - repeating lines 51
 - usage of 50-51
 - LOCATE command
 - changing current line pointer 31
 - command mode 141
 - current line and 32
 - description 140
 - examples 31-32
 - format 140
 - usage of 31-32
 - View Mode 141
 - locating data 69,124
 - logging off of VSPC 61,149
 - logging on to VSPC 34,180
 - logical records, file status information 156
 - logoff
 - CONTINUE file and 61,123
 - definition 188
 - description 149
 - examples 61
 - undelivered messages and 170
 - usage 61
 - logon
 - definition 188
 - description 180
 - examples 34,36
 - general description 17,18
 - message modes in effect 143
 - passwords defined for 150
 - PUNCH OFF in effect 155
 - QUERY obtains time in use 162
 - SHARE characteristic in effect 170
 - time limit for 181
 - TRANSLATE convention in effect 178
 - logon command
 - description 180
 - examples 34,36
 - format 180
 - usage 34
 - logon message
 - description 180
 - used in examples 34,36
 - logon password
 - changing 53,149
 - creating 149
 - definition 18
 - detailed description 149
 - general description 18
 - removing 149
 - replacing 149
 - lowercase
 - in examples 30
 - translation to capitals 58,59,178
- ## M
- managing a terminal session 36-45
 - manual setup of CPT-TWX terminal
 - KEY command and 129
 - PUNCH command and 155
 - TAPE command and 177
 - margins, not affected by LINESIZE command 134
 - maximum
 - increment in renumbering 164
 - job size in OS/V51 175
 - line length 108
 - logon delay time 181
 - message text length 169
 - number of tabs allowed 176
 - profiles sizes displayed 161
 - SUBMIT filenames 174
 - tab positions 176
 - MERGE command
 - description 141
 - examples 74
 - format 141
 - PROTECT command and 153
 - usage of 74
 - when password required 150
 - message
 - definition 188
 - display by STORE command 173
 - logoff 61
 - logon 34,180
 - MESSAGE command controls
 - display 62,143
 - rules for text 169
 - SEND command transmits 62,169
 - message area 47,48
 - MESSAGE command
 - description 143
 - examples 62
 - format 143
 - usage of 62
 - when blocks SEND command 170
 - message commands, general description 19
 - message header
 - definition 189
 - examples 63
 - format 63
 - in AID command 98
 - when omitted 62,143
 - when printed 62,143
 - messages
 - at startup after unexpected ending 60

- at unexpected ending 61
- batch job status 172
- batch job submission 175
- before input 35
- compiler 83
- explanations 42,63,108,185
- logoff 61
- logon 34,180
- receiving 62,143
- sending 62,169
- when undelivered 170
- minimum (*see* maximum)
- minus sign (-)
 - in command continuation 104
 - valid in message text 169
- mm/dd/yy (*see* date field)
- MORE...PRESS CLEAR (3270 terminal) 37
- MOVE command
 - description 144
 - examples 65
 - format 144
 - line renumbering and 65
 - usage of 65

N

- NAME command
 - description 145
 - examples 71
 - format 145
 - specifies passwords 150
 - usage of 71
- name field in filename
 - definition 23
 - description 122
 - QUERY LIBRARY command and 159
 - syntax of 23,122
- naming a workspace 71,145,173
- NEWLINE command
 - description 144
 - examples 55
 - format 144
 - usage of 55
- node-name, in HARDCOPY command 126
- NOREAD characteristic
 - file status information 156
 - LIST command display and 135
 - LOAD command execution and 136
 - MERGE command execution and 141
 - PROTECT command sets 153
 - summary of actions 131
- NOTEXT operand, description 54
- NOWRITE characteristic
 - file status information 156
 - PROTECT command sets 153
 - summary of actions 131
- null line, definition 131
- NUMBER command
 - asterisk in 147
 - CLEAR command and 115
 - description 147
 - examples 77
 - format 147

- LOAD job entry command and 136
- query NUMBER example 59
- service program statements and 147
- when password required 150
- numeric characters
 - in message text 169
 - in name field of filename 23,122
 - in passwords 150

O

- object program
 - definition 189
 - description 83
 - execution of 83
 - QUERY PROFILE and 161
 - RUN command executes 85,166
 - SUBMIT command restriction 174
- object program file
 - definition 189
 - detailed description 121
 - file status information 156
 - general description 24
 - LOAD command invalid 136
 - PROTECT NOREAD invalid 153
 - record number omitted in file status 156
 - STORE command creates 84,173
- object time
 - definition 189
 - description 83
- obtaining a file 74,136
- obtaining library directories 81,158
- OFF command
 - CONTINUE file and 123
 - description 149
 - examples 61
 - format 149
 - usage of 61
- OFF invalid as tablename 178
- ON LINE indicator to accept input 35
- online processing 15
- open message receipt 143
- operand
 - abbreviations of 20,104,105
 - definition 189
 - description 20,101
 - format notation for 105,106
 - format order 105
 - keyword
 - definition 20,104
 - overriding 104
 - punctuation in 20,106
 - purpose of 20
 - variable, definition 20,101
- operator message, receipt of 62,143
- operator, message recipient 169
- options, in formats 105,106
- order of processing, local commands 51
- organization of manual 3
- OS/VS1 considerations
 - batch job requirements 175
 - CANCEL command 97,112
 - LOAD command 94,136,137
 - ROUTE command 95-96

- SCRATCH command 98
- SUBMIT command 175
- OS/VS2 considerations
 - batch job requirements 175
 - CANCEL command 97,112
 - in jobnames 128
 - LOAD command 94,136,137
 - ROUTE command 95
 - SCRATCH command 98
 - SUBMIT command 175
- output
 - batch job
 - alternate destinations 95,165
 - erasing 97,112,168
 - retrieving 94,136
 - produced by RUN command 166
 - tab formatting of 55,176
 - output area, description 41
 - output characters, invalid 36,38
 - output class
 - batch job terminal output and 175
 - ROUTE command
 - specification 165
 - output file, PROTECT NOWRITE invalid 154
 - output line, PUNCH command controls 155
- overlap of lines
 - INPUT command and 43,126
 - MERGE command and 141
 - MOVE command and 144
 - RENUMBER command and 164
- overlay of file in workspace 75,141
- overriding keyword operands 103
- overview of VSPC 15-26
- overwriting of files, SAVE command prevents 167
- owner of file (*see* file owner)

P

- paper tape
 - PUNCH command controls output 155
 - TAPE command specifies input 177
- parentheses (())
 - in formats 106
 - used in operands 20
 - valid in message text 169
- partial job output (*see* output, batch job)
- password
 - (*see also* filename password and logon password)
 - access control feature 25
 - definition 189
 - detailed description 150
 - display suppressed 150
 - examples 25,35,78,89
 - filename 25
 - general description 16
 - logon 25
 - OFF continue and 149
 - omission in PURGE command 156
 - optional in filename 123
 - profile defines logon 149-150

PROTECT command
specifies 78,153
PROTECT WRITE can omit 154
specification of 150
syntax 25,150
unneded treated as incorrect 150
used with filename 23
VSPC command and 180
PASSWORD command
description 149-150
examples 53
format 149-150
usage of 53
PA1 key 37,38
PA2 key 38
period (.) valid in message text 169
PF key 38
PFKEY command
description 152
display settings 40
examples 38
format 152
usage of 38
plus sign (+)
in command continuation 104
valid in message text 169
pointer, current line (*see* current line
pointer)
positional operands
(*see also* operands)
in formats 105
POWER/V5 job statement, SUBMIT
command and 175
preface 3
print suppression of line
numbers 43,127
printer pages, QUERY commands
obtains totals 160
printer services 40
printing a workspace 72,135
printing data lines
CHANGE command and 68,113
FIND command and 69,124
LIST command and 72,135
printing file information 81,156
printing line numbers 43,127
printing, requests for 72,126,135
private library
access control summary 131
definition 189
detailed description 130
general description 23
library number references 122
SHARE command
considerations 170
STORE command and 173
privilege, profile, displayed by
QUERY PROFILE 161
PROCEED indicator to accept
input 35
processing batch jobs 90-96,97
processing command lists 87-90
processing CONTINUE file
prohibited 124
processing, offline 15
processing, online 15
processor, definition 189

processor-name attribute
(*see also* foreground processors)
definition 18
ENTER command 83-84,117
foreground processors
and 83-84,117
QUERY LIBRARY command
and 159
processor-names for IBM program
products 117
profile (*see* user profile)
program access key (*see* PA1 key or
PA2 key)
program function key (*see* PF key)
program product
definition 189
list of processor-names for 117
programming considerations for job
entry 91,92
programming language,
definition 189
programs, as data 21
project library
access control summary 131
definition 189
detailed description 130
file status information 156
general description 24
QUERY command and 156,158
SHARE command and 79,170
STORE command and 173
used in examples 30
project library files, library number
references 122
project library number
displayed by QUERY profile 161
zero (0) valid as
reference 71,130,158
prompt
by AID 98,99,107,108
definition 189
for passwords 151,180
PROTECT command
CLEAR command and 115
combining operands in 79
CONTINUE file disallowed 124
description 153
examples 78,123
file status information 156
format 153
LIST command execution and 135
LOAD command and 136
MERGE command and 141
password specification
through 150
PURGE command and 156
summary of actions 131
usage of 82
when password required 150
protection of files (*see* file protection)
public library
access control summary 131
definition 189
detailed description 131
file status information 156
general description 24
library number references 122

QUERY command and 156,158
SHARE command ignored 79,170
STORE command and 173
used in examples 30
publications references 3
PUNCH command, description and
format 155
punctuation in operands,
description 20,106
PURGE command
description 156
examples 82
format 156
usage of 82
purging of messages at logoff 169

Q

QUERY command
ACCOUNT example 160
description 156-162
format 156,158,160
NUMBER example 59
PROFILE example 60,162
TABSET example 55
TRANSLATE example 59
WORKSPACE example 162
QUERY FILE command to display file
status
description 156
example 81
format 156
QUERY LIBRARY command to
display directory
description 158
examples 19,81
format 158
usage of 81
querying batch job execution 93,172
question mark in AID
commands 64,99,107
quotation mark (')
CHANGE command and 113
encloses message text 169
FIND command and 124
in formats 106
used in operands 20,106
valid in message text 62,169

R

range of lines
CHANGE command and 113
COPY command and 66
DELETE command and 116
FIND command and 124
MOVE command and 144
read only access to
files 77,78,131,153,170
rearranging lines of data 65,144
receiving messages 62,143
record
batch job format 174
direct file format 121
excess characters cause error 174
object program files 121
sequential file format 122
record size

- batch jobs 174
- direct file rules 121
- file status information 156
- sequential file rules 122
- records, number of in file status information 156
- reformatting lines 67
- RELEASE command
 - description 163
 - examples 80
 - format 163
 - usage of 80
- releasing FORTRAN files 86,125
- remote job entry facilities (*see* job entry facilities)
- remote workstation, ROUTE command specifies 96,165
- removal of
 - filename password 150,153
 - files 82,156
 - logon password 151
 - PROTECT command restrictions 78,153
 - session tabs 176
- RENUMBER command
 - description 164
 - examples 70
 - format 164
 - usage of 70
- renumbering lines
 - entire workspace 70,164
 - MERGE command 74,141
 - MOVE command 65,144
 - part of workspace 70,164
- repeating lines, local command 51
- replacement of
 - data 65,113
 - file password 150,153
 - lines during data entry 43,44
 - logon password 35,150
 - object program by STORE command 173
- response to successful line entry 132
- restricting file access (*see* file protection)
- retrieving CONTINUE file 123
- right margin sequence
 - numbers 148,156
- ROUTE job entry command
 - description 165
 - examples 95
 - format 165
 - restriction on output deletion 170
 - usage of 95
- RUBOUT character, PUNCH ON appends 155
- rules for access
 - private libraries 23,130
 - project libraries 24,130
 - public libraries 24,131
- RUN command
 - command list and 87-90
 - current source program and 85
 - description 166
 - examples 85,89,90
 - file status information 156
 - format 166

- object program and 85
- PROTECT command and 153
- saved source program and 86
- usage of 85,87,87-90
- when password required 153

S

- S/S mode (*see* start-stop mode)
- SAVE command
 - command lists and 84
 - CONTINUE file and 124
 - description 167
 - examples 72
 - file status information 156
 - foreground processors and 84
 - format 167
 - password specification 150
 - PROTECT command restriction 154
 - usage of 72
- saving workspace contents
 - automatically 60,124
 - using OFF command 149
 - using SAVE command 72,84,167
- SCRATCH job entry command
 - description 168
 - examples 98
 - format 168
 - usage of 98
- SDLC (*see* synchronous data link control)
- search workspace for data (*see* FIND command)
- SEND command
 - description 169
 - examples 62
 - format 169
 - usage of 62
 - when unsuccessful 170
- sending and receiving messages 62,143,169
- separators
 - examples 21
 - in commands 21,103
 - in filenames 123
 - multiple recognized as one 21
- sequence numbers
 - assigning 77,147
 - definition 189
 - file status information 156
 - omission of 148
 - SUBMIT command defaults 174
- sequential file
 - definition 189
 - detailed description 122
 - FILE command and 121
 - file status information 156
 - general description 24
 - NUMBER command and 147
 - QUERY command and 156,158
 - sequence numbering for 147
- sequential undefined file
 - description 122
 - file status information 156
 - FILE command and 119
 - QUERY command and 156,158
- service program control statements, and NUMBER command 148
- session
 - definition 189
 - ending 61,149
 - starting 33,180
- SHARE command
 - CLEAR command and 115
 - CONTINUE file disallowed 124
 - description 170
 - examples 79
 - file status information 156
 - format 170
 - summary of actions 131
 - usage of 79
 - when password required 150
- shared storage maximums displayed by QUERY PROFILE 161
- sharing files (*see* SHARE command)
- signaling attention
 - Cancels output 36
 - character correction and 35
 - during STORE command execution 173
 - ends automatic line numbering 45
 - escape from input 36
 - interrupts job submission 93,175
 - line deletion and 35
 - system communication 36
 - WAIT message mode and 143
- signing off of VSPC (*see* OFF command)
- signing on to VSPC (*see* VSPC (logon) command)
- single line
 - CHANGE command and 65,113
 - COPY command and 66
 - DELETE command and 64,116
 - FIND command and 69,124
 - MOVE command and 144
- single quotation mark ('')
 - CHANGE command usage 113
 - FIND command 124
 - message usage 62,169
 - operand usage 20,106
- slash (/)
 - in filename 123
 - in formats 106
 - valid in message text 169
- MSG data set, LOAD command and 94,136
- source creation time 83
- source program
 - creation of 83
 - definition 189
 - description of 83
 - placed in library by SAVE command 84
 - retrieval 136
 - RUN command processes 85,166
 - size allowed, QUERY PROFILE and 161
 - STORE command compiles 84,173
- source time definition 189
- space (*see* blank)
- special character

- definition 189
- in formats 106
- in message text 169
- in name field 122
- special file CONTINUE 123
- SPLIT command
 - description 171
 - examples 67
 - format 171
 - usage of 67
- start-stop (S/S) mode
 - definition 189
 - logon command for terminals 34
- status information for a file 156
- STATUS job control command
 - description 172
 - examples 94
 - format 172
 - usage of 93
- STORE command
 - CONTINUE file disallowed 124
 - description 173
 - examples 84
 - file status information 156
 - format 173
 - password specification 150
 - PROTECT command
 - restriction 154
 - usage of 84
- string, definition 189
- string operand 166,173
- stroke (/) (*see* slash)
- SUBMIT job entry command
 - CONTINUE file disallowed 124
 - description 174
 - examples 92
 - format 174
 - PROTECT command
 - restriction 154
 - usage of 92
 - when password required 150
- submitting batch jobs 92,174
- superimposing files (*see* MERGE command)
- suppression of
 - line number printing 43,126
 - message header display 62,143
 - message display 62,143
- switched line, procedure for using 33
- synchronous data link control (SDLC)
 - definition 189
 - logon command for terminals 34
- syntax of
 - commands 19,20,103,104
 - filenames 23,122
- system administrator, definition 189
- system logon message (*see* logon message)

T

- tablist, TABSET command
 - specifies 55,176
- tabs
 - QUERY command checks 59,160
 - separators 21,103
 - setting 55,176
 - warning on matching 55
- TABSET command
 - column tab setting 177
 - description 176
 - example 55
 - format 176
 - QUERY command and 59,160
 - usage of 59
 - View Mode 176
 - warning on use 59
- TAPE command
 - description and format 177
 - KEY command and 129
- tape, PUNCH command controls
 - use 155
- telephone line, procedure for
 - logon 33
- terminal
 - asterisk specifies 86,110
 - batch job output 136
 - definition 189
 - features 183
 - QUERY command obtains
 - accesses 160
 - usage 30-35
- terminal oriented commands, general
 - description 19
- terminal output, PUNCH command
 - and 155
- terminal session (*see* session)
- terminals, description of
 - CPT/TWX 30
 - features available 183
 - 1050 30
 - 2741 29
 - 3101 27
 - 3270 27
 - 3767 27
 - 3770 29
- TEXT command
 - description 178
 - examples 53
 - format 178
 - usage of 53
- TEXT operand, description 53
- time
 - in logoff message 61
 - in logon message 34,180
- time limit for logon 180
- TRANSLATE command
 - description 178
 - examples 58
 - format 178
 - passwords and 150
 - QUERY command and 58
 - usage of 58
- translation convention, QUERY command checks 60,160

- translation of characters in
 - passwords 25,150
- transmitting messages (*see* SEND command)
- types of file in status information 156

U

- uncontrolled endings, CONTINUE file
 - and 61,124
- undefined file
 - definition 189
 - detailed description 122
 - FILE command and 119
 - file status information 156
 - general description 24
 - QUERY command and 156,158
- unit-number
 - ALLOCATE command
 - operand 86,110
 - definition 189
 - FREE command operand 86,125
 - unlocked keyboard, accepts input 35
 - unneded password, treated as
 - incorrect 151
- update of files
 - FILE command and 119
 - SAVE command and 167
- uppercase letters (*see* capital letters)
- upshift 2 character (@)
 - ignored for non start-stop terminals 34
 - required for start-stop terminals 34
 - valid in name field 122
 - VSPC command usage 180
- user library
 - detailed description 130
 - general description 23,24
 - usage of 74-82
- user number
 - definition 17,189
 - file status information 156
 - general description 15
 - message destination 62,169
 - QUERY PROFILE and 161
 - VSPC command operand 34,180
- user profile
 - creating logon password for 149
 - definition 189
 - description 17,18
 - display by QUERY
 - PROFILE 59,161
 - example 30
 - QUERY PROFILE note 161
 - removing logon password
 - for 53,149
 - replacing logon password
 - for 53,149
 - user library creation and 130
 - usernum (*see* user number)

V

variable operand
 definition 20,189
 in formats 103

VIEW command
 description 179
 examples 47
 format 179
 usage of 47

View Mode
 definition 189
 description 46
 line entry 132
 line length message 48
 processing procedure 25,133

VS APL for VSPC program product,
 description 82

VS BASIC program product
 batch processing and 91,174
 description of 82

VS Personal Computing (VSPC) (*see*
 VSPC)

VSAM (virtual storage access
 method), definition 189

VSAM files, external 25,122

VSPC (VS Personal Computing),
 definition 189

VSPC administrator, definition 189

VSPC AID
 command format 107
 definition 190
 description 19,107
 example 64,98,99

VSPC command functions list 102

VSPC concepts 15

VSPC facilities 15-26

VSPC FORTRAN
 ALLOCATE command
 and 86,110
 batch processing and 91
 command list processing and 89
 description of 87
 FREE command 86,125

VSPC general user's
 dictionary 101-181

VSPC libraries
 (*see also* libraries)
 definition 16,190
 detailed description 130
 general description 23,24
 usage of 74-82

VSPC (logon) command
 description 180
 examples 34
 format 180
 usage of 34

VSPC messages
 accepting input 35
 explanation by AID 64
 headers identify 63
 logoff 61
 logon 34,180
 startup after unexpected ending 61
 unexpected ending 61

VSPC operator as message
 recipient 169

VSPC overview
 description 15-16
 illustration of 17

VSPC publications available 3

VSPC terminals (*see* terminals,
 description of)

VTAM (virtual telecommunications
 access method),
 definition 190

VTAM node-name, HARDCOPY
 command operand 126

W

waiting for messages 143

warning on
 CPT-TWX setup 129,177
 linesize specification 54,134
 MERGE command files 142
 NAME command files 145
 tab list specification 55

work station, remote, specified by
 ROUTE command 96,165

workspace
 changing characters in 67,113
 changing numbering of 70,164
 contents after CLEAR command
 execution 73,115
 definition 190
 deleting lines in 64,116
 display area 47,48
 display of 72,135
 erasing completely 73,135
 erasing partly 73
 finding data in 69,124
 general description 16
 implicit PROTECT
 characteristics 154
 length displayed by QUERY
 FILE 156
 line number order in 42
 moving data in 65,144
 naming of 71,145
 placed in file 72,167
 PROTECT command and 153
 rearranging lines in 65,144
 renumbering of 70,164
 replacing data in 65,113
 RUN command executes 89,166
 saving contents of 72,167
 sequence numbering for 76,147
 SHARE command
 specification 170
 size displayed by QUERY
 WORKSPACE 162
 SUBMIT command
 specification 92,174
 unchanged by STORE
 command 173
 usage obtained by QUERY
 command 160

workspace attribute (*see* attribute)

workspace contents
 assumed in examples 30
 saving automatically 60
 when deleted or saved 60,149

workspace oriented commands,
 general description 19

workspace password, OFF
 CONTINUE and 149

writing source programs 84

wrong passwords, prompting and 151

X

XOFF character, PUNCH ON
 appends 155

XON key in input line 177

Z

zero (0) as project library
 number 71,130,158

Numeric

0 (zero) as project library
 number 71,130,158

0 through 9 in name field 23,122

2 character, upshift, in VSPC
 command 34,180

10 as default line increment
 INPUT command 63,126
 MERGE command 141
 MOVE command 144
 RENUMBER command 164

1050 terminal
 default line length 134
 description 30
 features chart 183
 restriction on input 35

2741 terminal
 default line length 134
 description 29
 features chart 183
 logon command for 34
 restriction on input 35

3101 terminal
 default line length 134
 description 27
 features chart 183
 logon command for 34
 restriction on input 35

3270 terminal
 character correction 37
 character deletion 37
 character insertion 37
 default line length 134
 description 27
 ENTER INPUT 38
 features chart 183
 illustration of keyboard 28
 illustration of screen format 29
 line correction 37
 logon command for 34
 MORE...PRESS CLEAR 37
 PA1 key 37,38
 PA2 key 38
 PF keys 38
 requesting printing 126
 used for examples 30
 valid for HARDCOPY
 command 126

3284/3286/3288 printer, HARDCOPY
 command requests 126

3767 terminal

default line length 134
description 27
features chart 183
illustration of layout 28
logon commands for 34
used for examples 30

3770 terminal

default line length 134
description 29
features chart 183
illustration of layout 30
logon command for 34



This Newsletter No. SN20-9317
Date 29 Feb 1980

Base Publication No. SH20-9071-2
File No. S370-39

Prerequisite Newsletters None

VS Personal Computing (VSPC) General User's Guide and Command Language

© Copyright IBM Corp. 1976, 1978

This technical newsletter, a part of Release 2 of VSPC, Program Products 5740-XR5 (OS/VS1 VSPC) and 5740-XR6 (OS/VS2 MVS VSPC), provides replacement pages for the subject publication. These replacement pages remain in effect for subsequent releases unless specifically altered. Pages to be inserted and removed are:

Title page, edition notice	93-118.1 (105,106,109,110,115-118 courtesy pages;
5-13 (page 14 removed and not replaced)	118.1 added)
19,20	125-128
27,28	133-142
39,40	147-156 (149,150 courtesy pages)
47-60	163-183 (181 courtesy page)
83-86	191-203

Each technical change is marked by a vertical bar to the left of the change.

Summary of Amendments

Changes included in this newsletter are summarized under "Summary of Amendments" following the list of figures.

Note: Please file this cover letter at the back of the publication to provide a record of changes.

VS Personal Computing (VSPC)
General User's Guide and
Command Language
SH20-9071-2

**Reader's
Comment
Form**

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

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.

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

List TNLs here:

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL _____

Previous TNL _____

Previous TNL _____

Fold on two lines, tape, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

Reader's Comment Form

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:

**IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150**



Fold and Tape



**International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604**

**IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591**

**IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601**



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601