

MUSIC/SP

Version 5

Release 1

User's Reference Guide

Seventh Edition (April 1996)

This edition applies to Release 1 of Multi-User System for Interactive Computing / System Product (MUSIC/SP) Version 5, and to all releases of this product until otherwise indicated in new editions or Technical Newsletters. MUSIC/SP Version 5 is published and licensed by McGill Systems Inc.

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: MUSIC Product Group, McGill Systems Inc., 550 Sherbrooke St. West, Suite 1650, Montreal, Quebec, Canada H3A 1B9. Fax: (514) 398-4488.

About this Manual

This manual details the user components of the MUSIC/SP system. No one needs to be familiar with all the information in this publication, as it is intended for advanced users.

First time users of MUSIC/SP should read the publication *MUSIC/SP Guide for New Users*.

Users of Personal Computers should also consult the *MUSIC/SP Personal Computer Workstation User's Guide* for information on dial-in access. Additional information about Personal Computers is found in *Chapter 2. Workstations*.

What is MUSIC/SP?

MUSIC/SP (Multi-User System for Interactive Computing/System Product) is a multi-user, multi-function, interactive system complete with a collection of application programs, compiler interfaces, and utilities. The collective goal is to provide a high-performance, cost-effective, and manageable computing environment. This environment can include IBM Personal Computers connected to MUSIC/SP running on an IBM main-frame processor.

Running as a single virtual machine, MUSIC/SP can support over a hundred concurrent users performing such diverse activities as problem solving, program development, accessing the Internet, accessing a CWIS, file editing, personal computer support, electronic office functions, and job submission with output retrieval to such systems as MUSIC/SP, DOS/VSE, and CMS batch.

This manual describes the usage of the MUSIC features and facilities available to the general user. Information on workstations, command language, job processing, and processor usage is included.

Chapter Overview

Chapter 1. Introduction

Provides an introduction to the MUSIC/SP system. Key concepts of the command language and file system are given. Also included in this chapter, is information about the menu-driven facilities available with MUSIC/SP.

Chapter 2. Workstations

Discusses how to use the various types of workstations that connect to MUSIC. Control keys specific to a particular workstation are described. This chapter includes information on the various ways of connecting IBM Personal Computers to MUSIC.

Chapter 3. Using Batch

Discusses how to submit batch jobs to MUSIC batch and other systems.

Chapter 4. File System and I/O Interface

Discusses the Save Library and User Data Sets. Tape and disk usage is explained. The unit number convention is described. Usage of VSAM files under MUSIC is explained.

Chapter 5. MUSIC Commands

Discusses MUSIC command language. Syntax and examples are given for each command. (The REXX command executor is documented in *Chapter 8. Processors*.)

Chapter 6. MUSIC Job Control Statements

Discusses Job Control Statements. Syntax and examples are given for each statement.

Chapter 7. Using the Editor

Discusses how to change files with the Editor. Syntax and examples are given for the Editor commands. The Editor can also be used to search files for specific information without modifying them.

Chapter 8. Processors

Discusses each compiler and loader available to the MUSIC user. The linkage editor is also explained. Concepts of source, object, and load modules are given. The REXX command executor is documented in this chapter.

Chapter 9. System Subroutines

Discusses the many MUSIC system subroutines callable from high-level languages.

Chapter 10. Utilities

Discusses the system utilities that can be used to copy, list, backup, and restore a file. PROFILE (the utility to change your password, and so forth) is also described.

Appendix A. IIAS/IIPS

Describes the IBM Interactive Instructional Systems IIAS and IIPS.

Appendix B. LEARN Program

Describes the online courses available with MUSIC/SP.

Index

Useful in locating a specific item or topic. Can be used when tracking down a word or abbreviation that you are not familiar with.

MUSIC/SP Publications

The following is a list of all the current MUSIC/SP publications. These hardcopy publications can be ordered through the MUSIC Product Group. Online versions (softcopy) of the user publications can be accessed with the MUSIC/SP command called "MAN".

- *MUSIC/SP Administrator's Guide* (April 1996), describes how to install and operate MUSIC/SP.
- *MUSIC/SP Administrator's Reference* (April 1996), describes the internals of MUSIC/SP; utility programs and supervisory commands; gives detailed storage estimates; and documents console messages.
- *MUSIC/SP User's Reference Guide* (April 1996), describes how to use MUSIC/SP; its command language; terminal and batch set up; and job processing using the various language processors.
- *MUSIC/SP Guide for New Users* (April 1996), introduces new users to the use of MUSIC/SP via an IBM 3270-type workstation. It describes the FSI (Full Screen Interface) menu facility. New users learn how to use many programs on MUSIC/SP for such tasks as editing and running programs.
- *MUSIC/SP Office Applications Guide* (April 1996), describes the features of the TODO (Time, Office, and Documentation Organizer) facility. This includes the scheduling function, spell checking, and MUSIC/SCRIPT (text processing).
- *MUSIC/SP Mail and Conferencing Guide* (April 1996), describes electronic mail on MUSIC/SP. This includes Mail Profile, Mail Directory, using POP clients, and conferencing programs.
- *MUSIC/SP Internet Guide* (April 1996), describes the programs available on MUSIC/SP that provide communication between users through electronic conferencing and discussion lists. Emphasis is placed on access to the Internet with programs such as TELNET (logging on other computers), FTP (File Transfer Protocol), WEB (World-Wide Web), RN (Newsreader), and GOPHER (document search and retrieval protocol).
- *MUSIC/SP Campus-Wide Information Systems (CWIS) Guide* (April 1996), describes how to create and maintain a Campus-Wide Information System, Help facility, or Classified Ads facility; how to do full-text searching; and how to provide gopher access. MUSIC/SP's resources are used to provide online distribution of information to a wide audience.
- *MUSIC/SP Teacher's Guide* (April 1996), describes various MUSIC/SP facilities related to the academic environment. Emphasis is placed on communication between teacher and student and easy methods for learning how to use MUSIC applications.
- *MUSIC/SP Client/Server (MCS) Booklet* (April 1996) provides an overview of MCS. Full documentation is available on the MCS diskette.
- *MUSIC/SP Personal Computer Workstation User's Guide* (May 1994), describes the components of the Personal Computer Workstation (PCWS). It is intended for the novice or experienced user of a personal computer, who wishes to connect to MUSIC/SP or another host system. Note that documentation for *PCWS for Windows* is available on the PCWS diskette.

Trademarks

- MUSIC/SP, TCP3270 are trademarks of McGill University, Montreal, Canada
- IBM, Micro Channel, AS/400, Personal System/2, PS/1, PS/2, and AIX are registered trademarks of International Business Machines Corporation.
- SAA, ES/9370, ES/9000, PROFS, PR/SM, MVS/ESA, VM/ESA, PC-DOS, DisplayWrite, OfficeVision, and GDDM are trademarks of International Business Machines Corporation.
- DEC VT-100 is a registered trademark of Digital Equipment Corporation.
- MINITAB is a registered trademark of MINITAB Inc.
- SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc., Cary, NC, USA.
- UNIX is a trademark of AT&T Bell Laboratories.
- Eudora is a registered trademark of the University of Illinois Board of Trustees, licensed to QUALCOMM Inc. QUALCOMM is a registered trademark and registered service mark of QUALCOMM Inc.
- Other names may be trademarks or registered trademarks of their respective companies.

Chapter 1. Introduction

Introduction to Interactive Computing

An **interactive computing system** is a facility through which a number of users can concurrently interact with the resources of one computer. Typically the user is connected to the computer by a device called a *workstation*. A workstation could be a host-dependent terminal or a personal computer. In this guide the term *terminal* is often used to mean the same thing as workstation. Although, technically, a personal computer is not the same as a terminal, it simulates a terminal when it is attached to MUSIC/SP.

This manual describes how to use the MUSIC/SP system. (MUSIC/SP is short for Multi-User System for Interactive Computing/System Product.)

How to Get MUSIC to Work for You

The first step is to connect to MUSIC from your workstation. This can be done in a variety of ways depending on your type of workstation. Refer to *Chapter 2. Workstations* about how to connect to MUSIC.

Once you have signed on to MUSIC you can choose one of MUSIC's menu-driven facilities or use MUSIC's command language. Command language is divided into two categories: MUSIC commands and MUSIC Job Control Statements. MUSIC commands tell MUSIC which task you want to do next. For example, you can edit or print a file, invoke a program, or start a menu facility. MUSIC job control statements are included in your files. They tell MUSIC such things as: what computer language your program is written in, where the data is to be read from, and the maximum amount of time for your job. For details refer to *Chapter 5. MUSIC Commands* and *Chapter 6. MUSIC Job Control Statements*.

Online Information

MUSIC provides online information in the form of electronic bulletin boards, conferencing systems, and help facilities. The MUSIC Help Facility can be invoked by pressing F1 or entering the command HELP. Here, you can find out about all the programs available with MUSIC. Each program you use also provides online help (F1). All MUSIC manuals are online and are available through the text searching facility. Use the command "MAN" to access online manuals.

Files on MUSIC

The MUSIC system uses two different file structures. One type is called a *Save Library File* and is simply referred to as a "file". This file structure is used for most applications. Each file in your Save Library (directory) can hold up to 56 million characters of data.

The other type of file is called a *User Data Set* or *UDS* file. Although for most applications Save Library files are used, UDS files are of great use in special situations. For example, UDS files can be as large as an entire disk pack if required.

MUSIC has several commands to create, modify, list and purge (delete) files. Also, MUSIC offers a *flat* or *tree* structured file system. With a flat file system, all your files are stored in one directory. It is up to you to devise your own naming conventions through the use of prefixes and suffixes to make distinctions between groups. A tree structured system allows you to make different directories for each group of files. The commands are the same as for DOS (MD, CD, RD). Full details about MUSIC's file system can be found in

MUSIC/SP Editor

MUSIC has a powerful mechanism for creating and modifying files. It is the *Editor* program, invoked by the MUSIC command EDIT. The Editor has several modes of operation and provides a wide variety of commands for editing files. On most workstations, function (PF or F) keys are used to perform common editing functions. The Editor can search for a particular record based on its contents; perform global changes; move, copy and delete sections; and merge parts of one file to and from another file.

The Editor is fully programmable and can be incorporated into other programs. You can create your own macros and tailor the editor to suit your needs.

Processors

MUSIC supports many different processors (compilers and interpreters). See *Chapter 8 - Processors* for full details.

Compilers are used to translate programming languages into detailed instructions the computer can work on. The MUSIC system supports the operation of many compilers and others can be added. This manual discusses several different ones. It is possible that your MUSIC system has access to more than those documented here. It is also possible that some of those documented here are not available for your use. Check with your installation about the compilers that are available.

Interpreters such as APL, VS APL, and VS BASIC are similar in concept to compilers, though they are classified as subsystems because they include extensive capabilities beyond that of compiling.

The REXX command executor is documented in Chapter 8. This high-level language allows the execution of MUSIC commands and programs from within the REXX program.

The FSI (Full Screen Interface) menu facility offers many interfaces for processors, eliminating the need to learn job control language. See this topic below for more information.

System Subroutine Library

A program will often call a subroutine to perform a calculation or get some information. For example, you might want to take the square root of a number, so you call the SQRT subroutine from FORTRAN. Such common subroutines are stored in MUSIC's Subroutine Library.

MUSIC has a great number of additional subroutines available that you can call from your program. For example, you can call one that gives you the current time, another gives the current date, and so forth. *Chapter 9. System Subroutines* contains details of these routines.

Batch Processing

MUSIC is designed to handle interactive users as well as *batch* users. Batch, with its high speed reader and printer, can be used to run jobs when the interaction from a user is not necessary. The same general command language is used for batch as well as interactive usage.

Utility Programs

MUSIC includes a collection of programs already written to perform tasks such as sorting and archiving. Most of these programs diagnose errors right away and allow you to correct them immediately. *Chapter 10. Utilities* describes those programs used to perform tasks directly related to MUSIC.

Overview

Easy access to MUSIC programs can be done through menu (or panel) facilities. This menu approach reduces the need to remember commands and helps novice users to be productive immediately. Menus provide fill-in-the-blank interfaces for many programs to help you specify needed parameters.

MUSIC/SP offers four menu facilities for users and one menu facility for the system administrator. The system administrator facility provides step-by-step menus for installing, monitoring, maintaining, and tailoring the system. It is documented in the *MUSIC/SP Administrator's Guide*.

Each of the four user facilities combine several functions of the MUSIC/SP system. These menu facilities fall under the categories of general users, student computing, teaching applications, and office applications. They are as follows:

- FSI (Full-Screen Interface) - General Users
- CM (Course Management Facility) - Teacher Applications
- CI (Course Information) - Student Applications
- TODO (Time, Office, and Documentation Organizer) - Office Applications

MUSIC/SP provides the programming tools to support the creation and modification of menus. You can easily create your own environment on MUSIC/SP or create an environment tailored for a particular group of users. You can choose to have one of these facilities start automatically each time you sign on to MUSIC/SP. For information about creating and changing menus, refer to the *MUSIC/SP Campus-Wide Information Systems (CWIS) guide* and the program "TMENU" in *MUSIC/SP Office Applications Guide*.

Notes:

1. Your installation may provide additional menu facilities other than the ones described here.
2. As well as using menu facilities on MUSIC, you can use MCS (MUSIC/SP Client/Server) software. Common MUSIC functions (editing files, e-mail, etc) can be done on your personal computer with the use of MCS. For more information use the MUSIC command "HELP MCS".

Full Screen Interface (FSI)

The Full Screen Interface (FSI) allows you to access various components of the MUSIC system through a series of selection menus. You can specify FSI as the auto-program in your user profile if you wish to have it automatically start when you sign on to MUSIC. This interface is full described, with many examples, in the *MUSIC/SP Guide for New Users*. Help is provided for each screen (program) as it is presented.

To start this interface from command mode (*Go), enter:

```
FSI [item]
```

The *item* is optional and can be one of the highlighted topic names from the main menu of FSI (See Figure 1.1).

Throughout the interface the following standard function key definitions are used.

- F1 provides help on the function and usage of the screen currently being viewed.
- F3 returns to the previous screen without performing any operation. This can be used to exit from the interface if pressed from the main selection menu.
- ENTER performs the operations indicated on the screen.

Main Selection Screen of FSI

```

Help  End  Up    Down  Top  Bottom  Main  Scan  Find  Topic  Quit
-----Full Screen Interface for MUSIC----- Page 1/1
Command ==>

Place the cursor on an item and press ENTER or RETURN.

MUSIC tools:
  Mail           Electronic mail facility
  Programming    Compilers, processors, tutorials, etc
  CI             Course Information
  Internet       Internet access, news reader, gopher, etc
  More           Other general MUSIC tools

MUSIC files:
  FLIB *         Full Library Screen current directory
  FLIB           Filespec=> < pattern
  FUTIL         Other file related utilities

MUSIC environment:
  Help           General help and online documentation
  New Password   Change your password
  Defaults       FSI customization
  Profile        Profile utility and options
  Disconnect     Terminate your session and disconnect from MUSIC
  /Suggest       Make a suggestion or send a comment to support staff

F1=Help          F3=End          F9=Find          F12=Retrieve

```

Figure 1.1 - Main Selection Screen of FSI

Menu for Teachers (CM)

The Course Management Facility (CM) allows teachers to communicate with their class via the computer. Teachers use the CM command to manage the course material and the students use the CI command (Course Information) to access the information they have prepared.

The Course Management Facility basically does three things. It allows teachers to create and manage files and make them accessible to the class in the form of notes, assignments or a course outline. The files are kept in the teacher's library and are accessible to the students, allowing the students to read them but not change them.

The system also maintains a class mailing list. This list contains the userids and names of the students in the class. It is used to access the student files and to send mail to individuals or the entire class.

Teachers are able to manage the students computer resource allocations and change their passwords.

CM Main Menu

```
Help   End   Up    Down  Top   Bottom Main   Scan   Find   Topic Quit
-----Course Management for EL101----- Page 1/1
Command ==>
      1 *NEW* mail items and      0 replies waiting
TAB (or use arrow keys) to move cursor to a topic name & press ENTER

Managing Information:
MAIL           Receive and send electronic messages
POP            Update the pop file
OUTLINE        Update course outline
NOTES          Update course notes
ASSIGNMENTS    Assignments management
LIST           Class list management
DOCS           Online documentation

Managing Student Environment:
TAILOR         Tailor student menu with extra options
CI             Invoke CI as a student

Additional Tools:  CONFERENCES    Electronic CONFERences MANagement
                  FILES           View student files
                  AGENDA          Invoke TODO facility
                  TRANS$          Transfer funds and change passwords
F1=Help F2=Ask F3=End F7=Up F8=Dn F9=Fnd F10=Tp F11=Bt F12=Cur PA1=Quit
```

Figure 1.3 - CM Menu Display

Full details about this facility can be found in the *MUSIC/SP Teacher's Guide*.

CI (Course Information)

The Course Information menu is provided for students of teachers using the CM (Course Management) facility. The teacher provides information, such as course notes, assignments, and makes this available through CI. The following menu is a sample that students would work with:

```

Help   End   Up    Down  Top   Bottom Main   Scan   Find   Topic Quit
-----Course Information for CINFO----- Page 1/1
Command ==>
Date: 19May94 14:31:27                      Updated: 10May94 16:30
TAB (or use arrow keys) to move cursor to a topic name & press ENTER

General Information:
MAIL           Receive and send electronic messages
OUTLINE        Read course outline
NOTES          The course notes
ASSIGNMENTS    The assignments
DOCS           Online documentation
FSI            The menu interface for MUSIC

COGN           Conference about Cognitive Computing
CONFDAT        Conference about Databases
POLYSOLVE      Calculator
SN             SchoolNet access

F1=Help F2=Ask F3=End F7=Up F8=Dn F9=Fnd F10=Tp F11=Bt F12=Cur PA1=Quit

```

Figure 1.4 - CI Menu Display

Help is provided once the facility is invoked.

Time, Office, and Documentation Organizer (TODO)

The Time, Office, and Documentation Organizer Facility provides an integrated package tailored to the electronic office environment. This facility is called TODO for short.

To invoke TODO, type TODO from command mode (*Go); the following screen appears:

```

----- TIME, OFFICE, AND DOCUMENTATION ORGANIZER -----TODO
SELECT OPTION ==>_

                                TIME: 11:49 am
1  Schedules
2  Electronic Mail <option>      1989      FEBRUARY      1989
3  Telephone Log
4  Calculator <calc>             S    M    T    W    T    F    S
5  Spell Check document <option>                1    2
C  Create new <filename>         3    4    5    6    7    8    9
R  Revise <filename>             10   11   12   13   14   15   16
X  Execute SCRIPT <filename>     17   18   19   20   21   22   23
S  Submit SCRIPT <filename> <options> 24   25   26   27   28
L  List File Names <options> <pattern>
M  Schedule a Meeting <options>                                Day of year: 49
U  Utilities <option>

=====
F1:Help on Menu F2:Today's Reminders F3:Exit F6:Mail Waiting F12:Retrv

```

Figure 1.5 - TODO Menu Display

The TODO facility allows you to select a function from a menu list. Items include:

- Spelling check of a document or a single word. In document mode, the word and the surrounding lines are presented in an editor-like environment to allow correction.

Alternative spellings are provided for misspelled words. A 90,000+ word English language dictionary is included.

- A calendaring system allows you to maintain a personal daily schedule. You can authorize others to look at the calendar and allow certain specific users the ability to change items. The system maintains a record of who changed each item.

This facility can also be used to schedule conference rooms and equipment. Users can directly update the appropriate schedule without the need of sending messages to a room coordinator. Internally the facility uses only standard MUSIC files. A user's appointment data is placed in a separate file for each month to minimize the number of files involved.

- A monthly calendar can be displayed. The user can display future and past months as well as the current one. Reminders can be set keyed on a specific date or all Mondays, etc.
- A log of telephone calls and notes about each call can be maintained.
- An option to assist in the creation of letters and memos according to a user-modifiable style.

Consult the separate publication *MUSIC/SP Office Applications Guide* for full details.

Word Processing

MUSIC/SP supports three word processing programs. MUSIC/SCRIPT, IBM DisplayWrite/370, and Waterloo SCRIPT (WATCOM Products Inc.). MUSIC/SCRIPT is included with the MUSIC/SP system and is described in detail in *MUSIC/SP Office Applications Guide*.

DisplayWrite/370 is an optional IBM program that may be available at your installation. This program is a host-based text editor and formatter.

Waterloo SCRIPT is a powerful and versatile text formatter, written and distributed by the University of Waterloo. Some of the features of Waterloo SCRIPT include proportional spacing, the ability to generate boxes around text, footnotes, producing multiple columns of text on a single page, creating indexes, and hyphenation. Waterloo SCRIPT supports many popular output devices ranging from simple printing terminals to sophisticated laser printers.

MUSIC/SCRIPT

TODO includes MUSIC/SCRIPT as one of its components. MUSIC/SCRIPT is a set of text processing application programs that run on MUSIC.

The MUSIC/SCRIPT facility allows secretarial and administrative, as well as programming personnel to take advantage of the computer's resources for preparing, storing and producing final documents ready for mailing or publication. It is particularly useful for the preparation of letters or technical or legal documents that must be letter perfect or are subject to constant revision.

Corrections and revisions need only be made to the areas actually needing modification, thus saving the need to retype and recheck the unchanged areas. Furthermore, MUSIC/SCRIPT includes features that enable the user to identify those modified areas in the printout of the modified text. The actual modifications to the text are made with the MUSIC editor.

Special control words can be inserted into the input text files to control the format of the output. The output can be immediately displayed at your workstation, or sent to a printer.

The MUSIC/SCRIPT facility includes utility programs which can be of great assistance in the preparation of documents. These include contents and index creation programs.

MUSIC and You (Userids and Profiles)

What is a Userid?

When you connect to the system you must identify yourself by entering your userid and password. (Sometimes a userid is referred to as a sign-on code.) Each MUSIC installation may have thousands of authorized userids. Usually each user is assigned a different userid, though some installations may assign a single userid to be used by many users. MUSIC does not restrict the number of users that can be actively signed on with the same userid.

Each userid can be from 1 to 16 characters long and can optionally include a subcode of up to 8 characters. The userid, excluding any subcode, is used to identify the ownership of the files on the system, and is called the *ownership id*. Users can share the same files by having the same userid with different subcodes. Typically, userids do not have subcodes and each person has their own private files.

Password

Each userid is protected by a password. You must type in the correct password for the userid in order to be allowed to sign on to MUSIC. Passwords can be from 1 to 8 characters in length. It is good practice to use at least 6 characters and not to choose a person's first name as a password, as others may guess it. The password should be kept private between you and MUSIC. It is a good practice to change your password frequently (at least once a month) to protect the integrity of your userid. You can change your own password by using the PROFILE program that is discussed in *Chapter 10. Utilities*.

Time Limits

Each job that you run on MUSIC is subject to a time limit. This guards against a job going into a *loop* (running indefinitely) for hours when it should have finished in seconds. This limit also protects you from incurring the high usage charge for such jobs.

Time limits are given in *service units*. A service unit is equal to the amount of work done in one second elapsed time on a computer that processes a million instructions per second. (Installations have the option of changing this definition.)

A default time limit is assigned to each userid. This default time will be used if a job does not specifically supply one. You can specify a higher or a lower time limit for a job by way of the /SYS TIME= statement. For information about /SYS see *Chapter 6. MUSIC Job Control Statements*.

Fund Allocation

The installation may allocate a finite amount of funds to a userid. The amount of funds remaining is updated on a daily basis with processing unit time and connect time charges only. When there are no funds remaining, the user is prevented from signing on until more funds are allocated to that userid. Consult the description of the /ID command and the PROFILE program for ways of finding out the amount of funds remaining in your userid.

Disk Allocation

The installation may allocate a limit to the amount of disk space that you may use for storing files. A limit can also be set on the maximum size of each file. These limits are applied to individual userids. You can check these limits by using the PRINT command of the PROFILE program. (The PROFILE program is described in *Chapter 10. Utilities*.)

Your disk space allocation limits can be changed by the system administrator.

Userid Profile

As described above, each userid has a password, default time limits, funds and storage associated with it. Actually, there are many more items for each userid. This list of items is called the *User Profile*. The MUSIC system administrator initially sets up your profile when your userid is authorized. Various limits may be placed on your userid. Naturally, you are not allowed to change some of these limits. There are, however, many items that you can change in your profile. The following list highlights some of the items that are contained in your profile. Consult the writeup on the PROFILE program in *Chapter 10 - Utilities* of this manual for a description of how you can display and modify your own profile.

Usage Constraints Given in the Profile

- Time limit for batch jobs
- Time limit for prime-time jobs at your workstation
- Time limit for nonprime-time jobs at your workstation
- Batch access - yes or no
- Maximum User Data Set size that can be created
- Maximum fund allocation for this userid
- Maximum number of extra sessions
- Password changes are or are not allowed
- The ability to save files accessible to others can be allowed or disallowed
- The ability to save files in the common index (public files) can be allowed or disallowed

User Modifiable Items

- Userid Sign-on password
- Batch password
- Default printer location
- Whether implied EXEC feature is active or not
- Default job time
- Whether *In progress messages are to appear
- Input tab settings
- Output tab settings
- Default tab character on input
- Default backspace character

MUSIC System Overview

What does the computer do and what does MUSIC do? Even though you can view MUSIC and the computer as one entity, you might find it interesting to find out what MUSIC really is. This section provides a brief explanation of the internals of the MUSIC system. If you are not interested, skip the rest of Chapter 1.

MUSIC can be viewed as just a large program. Actually, MUSIC is one of a class of programs known as operating systems. (Some operating systems are also called *System Control Programs*.) Operating systems are distinguished from ordinary programs in that they can control an entire computer system by themselves and allow other programs to run under their control. Operating systems use special *supervisor* instructions to control the devices connected to the computer. Before describing what the MUSIC operating system does, a brief description of the physical hardware of the computer system is given. (Notice that the word *hardware* is used to describe the physical parts of a computer system while the word *software* is used to describe the programs that run on it.)

Hardware Components

The heart of the computer system is the Central Processing Unit (CPU). The processing unit gets all its instructions from main storage. Main storage can typically hold several million characters of information and it is generally contained in the same housing as the processing unit.

The processing unit and main storage use external devices to read in and write out information. These Input/Output units (I/O) are connected to the processing unit via channels. There may be many I/O devices sharing and competing for the use of a single channel.

Disk devices are used to extend the storage capacity of the computer system. A single disk can typically store between 300 and 3000 million characters of information depending on the type and model. The information stored on disk is only accessible once it has been read into main storage. The term *blocks* refers to the units of information read in from disk. These blocks are read into areas known as *buffers*. Blocks can also be written from main storage to disk. Buffers can exist almost anywhere in main storage, though on MUSIC most of the buffers are grouped together.

Each workstation is an I/O device, though it may be located thousands of miles away from the processing unit. Each workstation is connected to the channel via a *port* on a transmission control unit. There may be dozens or hundreds of ports available. Each time you use MUSIC you may be connected to a different port, but you needn't be aware of this.

Other I/O devices connected to the channel include magnetic tape and high speed printers, and an operator's console.

MUSIC System Tasks

MUSIC manages the network of I/O devices, channels, and main storage. It is concerned with space management in main and auxiliary storage. It maintains usage records for accounting purposes. User requests are monitored by MUSIC to protect against violations of system security such as attempts to read another user's private files.

Even though MUSIC is involved in a large number of activities, it still has time to individually respond to

each user's requests and handle them in an appropriate order relative to other users.

VM and MUSIC

The two operating systems, VM and MUSIC, can run on the same computer at the same time. VM is a special type of operating system designed to run other operating systems under it. With VM, a computer can be used for the MUSIC interactive work and can run the current administrative system at the same time without conversion effort. VM allows the installation to divide the computer's resources, such as the physical I/O devices and the processing time amongst the operating systems under it.

MUSIC running in a VM environment offers some facilities to users which are not available otherwise. For example, TCP/IP services. It is possible to create jobs using the advanced file and editing capabilities of MUSIC and then *submit* them via VM to any other operating system also running under VM. Output from jobs run on many other systems can be retrieved from MUSIC. In this way MUSIC can provide remote job entry (RJE) to the other operating systems that cannot handle workstations directly or that do not handle them in a convenient cost-effective manner. See the SUBMIT writeup in *Chapter 3. Using Batch* for further information.

MUSIC workstations can submit jobs directly to MUSIC's batch facility using VM. This allows jobs requiring disk or tape mounts or those producing large volumes of output, to be scheduled to run directly by users from the workstation. Furthermore, jobs can be submitted to be run overnight, allowing MUSIC to concentrate its efforts during prime time to those problems requiring immediate solution. See the SUBMIT writeup in *Chapter 3. Using Batch* for further information.

User Region

MUSIC sets aside an area to be used for user programs. In computer terms, one can say the user region is 256K, where *K* represents 1024 characters. (The term *byte* refers to the unit of storage that can hold one character. Typically a computer instruction or a number requires 4 bytes.)

Together with the user's program and data, this user region also contains I/O buffers and control information and a user-system interface module. This interface module is used to communicate requests of the high-level language into requests that are performed by the MUSIC system. This module is also used to present errors to the user in techniques dependent on which high-level language is used. The interface module may also contain some common subroutines used by most programs.

MUSIC supports region sizes larger than 256K. In fact they can be up to several million bytes in size. Check with your installation for the maximum size available to you. If this is too small, use of an *overlay structure* using the *linkage editor*, allows you to run even larger programs.

User Servicing Techniques

Since many users may want to run programs at the same time, the operating system must have techniques of temporarily stopping one job so as to let another job run. MUSIC uses three techniques to accomplish this: dispatching, time-slicing, and spooling.

The first technique allows another job to run while some other job is waiting for I/O to complete. This is done by a part of MUSIC called the *dispatcher*.

The other technique is called *time-slicing*. A time-slice is typically a small fraction of a second. After each

time-slice, MUSIC checks to see if another user should be serviced next. If so, a region switch is initiated to put the user's job into a dormant state and to activate the other. At a later time, the dormant job can be reactivated to allow the job to continue from where it left off. Active jobs requiring additional storage may cause the system to decide to *swap* a dormant job to disk.

Quite often a job does not go for its full time-slice such as in the following case. A job requires a read operation from a workstation, and it cannot continue until the read operation is done. It may have to wait several seconds or minutes for the user to respond. MUSIC will consider the time-slice is over in such a case, and try to service another user while waiting.

User programs always require the user region while they are running. However, MUSIC can perform many operations without the need for the user region. Examples include the INPUT mode of the Editor on ASCII workstations, the handling of output to the workstation and the initial check of commands which are typed in. MUSIC accomplishes these operations by using a special *spooling* system which allows the user to type in many lines and receive many output lines without the need of the user region. This spool system uses main storage to equalize the demands of the processing unit that can process millions of characters a second to those users who can type about 10 characters per second at best.

Even though the spool system, the dispatcher, and time-slices are fundamental to the operation of MUSIC, the user need not be aware of them, nor do the programs require special instructions to handle them. The beauty of MUSIC lies in the fact that the user need not be concerned with system operation. Application programmers can concentrate on writing and checking their programs; while users of MUSIC subsystems don't even have to know what programs or computers are all about.

Chapter 2. Workstations

Overview of Workstations

This chapter describes various types of workstations and how they are connected to MUSIC.

Basic Concepts

Most people use MUSIC through a typewriter-like device called a workstation or terminal. The term "workstation" and "terminal" are often used interchangeably, although there is a distinction. Workstation is a broader term referring to any type of keyboard device such as a personal computer or a 3270 terminal. Since PCs emulate terminals when they are connected to MUSIC, the word terminal is often used to refer to both types.

Basically, the workstation keyboard is used to transmit commands and data to the computer. After typing text using the alphanumeric keys, you must press the ENTER key to send the data to the computer. The ENTER key informs MUSIC that you have finished typing, and it is only at that point that it will process the data. On some workstations the ENTER key may be referred to as RETURN key.

If you make a typing mistake it can be corrected prior to pressing ENTER by using special local editing keys. Most workstations have a BACKSPACE key which allows you to backup to the error and retype the line from that point. The INSERT and DELETE keys allow you to insert and delete characters in the middle of the line.

A BREAK key is provided to let you interrupt the processing of a command and enter special *break time* commands to inquire about how much time has been used so far, or to skip some lines of output, or to cancel the current job entirely. (3270-type workstations use the PA1 key to provide the BREAK function).

3270 Architecture

The MUSIC system runs on computers that use IBM 3270 architecture. This means that any workstation connecting to MUSIC must be able to communicate in this 3270 environment.

There are many types of workstations that can be used on MUSIC. Many workstations emulate the characteristics of an IBM 3270 terminal and are referred to as "3270-type" workstations. This type of workstation is used most often with MUSIC and supports full-screen operation and function keys.

Detailed information on some specific workstations is given later in this chapter.

Connection to the Computer

All workstations have to be connected to MUSIC via some sort of wire. This connection may be in the form of a regular telephone line (workstation with a modem), or by a direct cable to the computer room.

Workstations which are connected to the computer through a direct cable are likely to be 3270-type terminals. Consult the description of "IBM 3270-Type Terminals" in this chapter for details on making the connection to the system for these types of workstations.

Once the connection to the mainframe is made, you should follow instructions at your workstation to get the MUSIC sign-on screen. If the sign-on messages do not appear, press the ENTER key to clear the screen and the messages should appear. (For specific details about the operation of a particular workstation, consult the

description of the various types of workstations in the later part of this chapter.)

Types of Connections

The following is an outline of the different types of connections supported by MUSIC. Later in this chapter, each type is covered in its own section to provide more details.

1. Using 3270-Type Terminals

The first major class of workstations is the IBM 3270 family of terminals. They are display terminals that use the EBCDIC encoding conventions. They connect via a coax line to a 3270 controller. Since these terminals are usually directly connected they offer a good data transfer rate. A number of different terminal models are part of the 3270 family. The IBM 3270/PC is a special one in that it can also operate as a stand-alone PC.

Most IBM host software is written to support 3270-type terminals. These operate in Full Screen mode, providing programs such as the MUSIC Editor, with the ability to define input and output areas anywhere on the screen.

Note: Many types of workstations can emulate 3270s and are referred to 3270-Type terminals. Each of the other types of connections below provide some method of emulating 3270s - thereby, taking advantage of full-screen mode of operation.

2. Using ASCII Terminals

The second major class consists of ASCII workstations. These can connect directly to the computer or by using a modem. Depending on the computer hardware at your installation, your workstation may operate in line mode or, if your site has a protocol converter, it can emulate a 3270 terminal. Your installation should be consulted for these details.

3. Using a Personal Computer with PCWS (or other communications package)

PCWS is a terminal emulation program distributed with MUSIC and designed to run on computers from the IBM PC, PS/1 and PS/2 families. This communications package allows your PC to be remotely connected using a modem. Designed specifically for communicating with MUSIC, it provides reliable file transfer and 3270 terminal emulation without a protocol converter.

PCWS also provides VT100 terminal support if your PC is connected via a protocol converter. The VT100 terminal type is widely used with many types of computer architecture.

Although using PCWS to communicate with MUSIC offers a number of advantages, PCs can also use other commercially available communications programs.

4. Using NET3270 for Personal Computers on a LAN

NET3270 is an optional product that can be purchased through the MUSIC Product Group. This software provides 3270 emulation to all IBM PC, PS/1 or PS/2 connected to a LAN (Local Area Network). The LAN hardware can be Token-Ring, Ethernet, or Arcnet.

5. Using the Internet

If your workstation is running TCP/IP software and has access to the Internet, you can sign-on to a MUSIC system. You can connect from your workstation or from another computer system. The command you use depends on the software - the most common commands are "TELNET" and "TN3270".

More details about each type of connection is given later in this chapter. This information serves only as a guide for operating the workstation and should not be used in deciding which workstation to get. Your installation can usually be of assistance in advising you about this matter. You should now read that section

pertaining to the type of workstation you intend to use. Alternately, you could try to find someone who would take a few minutes to explain the workstation operation to you.

Workstation Defaults

When you sign on, the system can automatically distinguish between 3270 and ASCII workstations and it will select the appropriate operating characteristics accordingly. Since not all workstations within each of classes are the same, sometimes additional information must be supplied indicating the exact model of workstation being used. This information can be specified using the *trmcsls* option of the */ID* command or the *TERM* option in your profile. Other profile options can also be used to set personal defaults for such things as input and output tab settings. These settings can also be dynamically changed by MUSIC commands during the time you are signed on to MUSIC. Refer to the writeup on the PROFILE program in *Chapter 10. Utilities* section of this manual for more information about how you can examine and change the profile defaults.

IBM 3270-type Terminals (Connection type 1)

Supported Models

MUSIC supports the 3178, 3179, 3180, 3277, 3278, and 3279 display terminals (80 and 132 characters per line models) of the IBM 3270 family when they are locally attached to the processing unit. It also supports the 3270 Personal Computer, Personal Computers with 3270 co-axial adapters, and terminals connected through 3270 protocol converters such as the 7171.

Remotely connected terminals can be supported to appear as though they are locally attached terminals to MUSIC. The software and hardware to do this is outside of the domain of MUSIC.

Special Keys

The ENTER key is used on the 3270-type terminal as an end of line signal. It is located on the bottom right of the main keyboard. The BACKSPACE key is used to correct typing errors you detect in a line before the ENTER key is pressed. There is no limit to the number of times you use the BACKSPACE key.

On 3178, 3180, 3278 or 3279 terminals, the ALT key to the right of the space bar has special significance. On some terminals, it must be held down while the PA1, PA2 or CLEAR keys are pressed.

The PA1 key is used to generate the break signal, which puts the terminal into attention mode, indicated by ****Attn**** in the lower right corner of the screen.

The PA2 key is used to clear the top portion of the screen. This key can be pressed in response to a **More . . .** status. With MUSIC, the PA2 key is also used during full-screen programs to request a Multi-Session function. Multi-Session is described later in this chapter. (On some keyboard arrangements, this key may be marked CNCL instead of PA2.)

The PA1 and PA2 keys on some keyboards perform a dual function depending on whether the SHIFT key is also pressed. For example, the PA2 key may have FIELD MARK also written on it. If this is so, then pressing this key may not result in the display clear operation you think it should. This could be the result of pressing the LOCK key accidentally causing the keyboard to be locked in upper case. Pressing the SHIFT key solves this problem.

The DUP, TEST, and FIELD MARK functions are generally not used on MUSIC.

MUSIC supports the APL ON/OFF key. Refer to the VS APL description for details on this usage.

Entry Assist Feature

3178, 3278 and 3279 terminals can have an entry assist capability available on them. This assist provides many desirable features such as automatic wordwrap, column tabbing, end-of-line signal and cursor positioning information. The feature is enabled by pressing the DOC ON OFF key located below the key marked ATTN. Consult your installation for information on how to use this feature.

Power On & Display Intensity

3277

Turn the terminal on by gently pulling the power-on knob located at the bottom left of the screen. You can rotate the front of this knob for the proper intensity. It is best not to have the intensity too high. The intensity of the characters that you type in can be made less than the rest of the display. This is controlled by the back half of this knob.

3178/3278/3279

Turn the terminal on by pushing the red switch located at the left of the screen. The two dials at the right of the screen control the intensity and contrast of the displayed characters. The two switches on the right of the screen should be set in their "A,a" and "Normal" positions.

Sign on to MUSIC

Once the terminal is powered on it will usually display the word MUSIC/SP in block letters. Press the ENTER key to get the sign-on message.

If the terminal displays the characters VM then you must press the ENTER, type "DIAL MUSIC" followed by the ENTER or PA2 key.

If neither MUSIC or VM is displayed, press the CLEAR key.

Note: For purposes of the Editor, it is important for MUSIC to know the actual number of PF Keys (program function keys) on the terminal being used. At sign-on time, if the /ID line is entered by pressing the ENTER key, MUSIC assumes 24 PF Keys if the terminal appears as an IBM 3178, 3278 or 3279. Otherwise (e.g., IBM 3277), MUSIC assumes 12 PF Keys. The user can override these assumptions by pressing the highest numbered PF Key to enter the /ID line at sign-on time, rather than the ENTER key. The highest numbered PF Key is PF12 or PF24 and is always located at the extreme lower right hand corner of the keyboard. Refer to the description of the /ID command in *Chapter 5. MUSIC Commands* or to the "Editor Full-Screen Mode" description in *Chapter 7. Using the Editor* for more information.

Screen Format

MUSIC uses the top lines of the screen as an output area. You normally type information in an input area located one line from the bottom of the screen. A line formed of dashes and the letter T will appear directly above this area. The location of the T's correspond to your TAB settings and you can change their location by the /TABIN command. The TAB key on the 3270 is not normally used for any tabbing effect.

A MUSIC state message appears at the bottom right of the screen, for example, "Reading".

A special cursor character that looks like an underscore (_) shows the location where your next typed character will appear.

Screen Control

The display screen output area fills up starting with the top line of the screen and continues down until all the lines in the output area are filled. Lines greater than 80 characters will appear on two lines. The `More...` message will appear when MUSIC wants to display a line and no more space is left in the display area. Press the ENTER key to clear the display area and allow MUSIC to continue its output. Some programs may issue commands to start at the top of the screen without completely filling the display area. The `More...` message will be displayed in this case as well.

MUSIC Screen States

The MUSIC screen state message at the lower right can be one of the following:

Reading	MUSIC is waiting for the user to enter some information.
Attn	The terminal is in <i>BREAK</i> status waiting for the user to type some command. This status is caused by pressing the PA1 key.
Working	MUSIC is working on your last request and no output is currently available.
More...	MUSIC is waiting for the user to press the ENTER key to clear the output display area, since it has more output waiting.
Writing	MUSIC is currently writing information on your screen. You may never see this message as MUSIC can normally write to the screen faster than your eyes can detect. This message will, however, appear when the system is processing your /CANCEL request.

Advanced Typing Techniques

This terminal allows for advanced character editing of input information before it is passed to MUSIC. You will notice that the BACKSPACE key immediately corrects any typing error.

You are not able to change anything outside your 80 character input area. Attempts to do so will cause the INPUT INHIBITED screen status indicator to come on. The RESET key can be used to turn this indicator off. Then, move your cursor back into the input area with an arrow key.

The cluster of 4 keys with arrows pointing in different directions can be used to move the cursor anywhere on the screen. The DEL key can be used to delete the character in your input area pointed to by the cursor and results in the rest of the line being shifted one character to the left.

The INS MODE key can be used to insert some characters at the point identified by the cursor. You must press the RESET key when you are finished the insert operation.

Pointing the cursor to any line in the output area (including the TAB line), followed by pressing the ENTER key, will cause the line to be copied to your input area. This is called *line call down*. Once in your input area, you can modify it by using the character editing facilities of the terminal. Press ENTER to accept this line as input. (Note that this *line call down* feature copies the line as **displayed**. Thus if the original line had some unprintable characters in it, they will not appear in the copy brought down to your input area.) Line call-down cannot be used on 3270s connected via VM Remote Dial.

The RETRIEVE function (PF12) retrieves the last command entered and scans back through these

commands each time PF12 is pressed. These commands are displayed in the input area and can be modified and re-entered. (See the /DEFINE command in *Chapter 5. MUSIC Commands* for additional information.

The ERASE INPUT key can be used to completely clear the input area. The ERASE EOF will clear the input area from the cursor position on to the end of the input area.

MUSIC allows the user to start typing the next line of information before it has finished processing the first. Thus while your terminal is in Working or More status, you can type in the next line. You can even press the ENTER key if you are sure you want this line next.

Notes

1. A BREAK request can be signalled when the terminal is in More status by pressing the PA1 key. This will first clear the screen and then put the terminal in ATTN (break) mode.

While the terminal is in attention mode (ATTN), you can press PA1 to skip the remaining output. This is equivalent to entering the command /SKIP ALL. Thus, to skip the remaining output when in More status, press PA1 twice (the first one enters ATTN mode and the second one does the skip).

2. The space bar or the special cursor movement keys can be used to move the cursor to the right. They are not equivalent. The cursor movement key does not put in blanks as it moves. Consequently if you extend your input line by using the cursor keys you will find that the apparent blank characters are not really there when you press the ENTER key.
3. All lower case characters will be displayed as their upper case equivalents on some 3270 terminals, but the terminals still send MUSIC the lower case characters. This presents no problem to the user unless the user is working with a file that was created with TEXT LC or equivalent. Such is the case when the file is to be used by the MUSIC/SCRIPT subsystem.
4. Pressing the CLEAR key when you turn on your terminal and the previous user has not signed off, enables MUSIC to initialize your display. If your display is not initialized in this manner, the output will not be formatted correctly.

3270/PC and other Co-Axial Connected PCs

MUSIC supports PCs connected via co-axial cable. The IBM 3270 Personal Computer uses that kind of connection. PCs connected in this way essentially function as regular 3270 terminals. The file transfer can be provided using the IBM TSO file transfer program. The procedure for transferring a file is as follows:

SEND : Transferring a file from the PC to MUSIC.

1. Sign on to your MUSIC system from a host session in 3270 mode.
2. Make sure the screen is cleared. You can either press the CLEAR or PA2 key to do this. Failure to do this will cause an error message to appear. Follow instructions at the end to recover from this situation.
3. When running on a 3270/PC you must press the JUMP key located on the left hand side of the keyboard to go to the PC session. If you have windows defined, the window must be expanded to full size.
4. Issue the following command when the DOS prompt appears:

```
SEND fromname x:toname  [ASCII] [CRLF] [APPEND] [RECFM(V|VC|F|FC)]
                        [LRECL(n)] [SPACE(n)]
```

where:

fromname	is the name of the file being sent to MUSIC. This name can include a drive specification and a path. (Refer to <i>Specifying the Path to a File</i> in your IBM Personal Computer DOS manual for further information.) This is a required option.
toname	is the name of the MUSIC file where the transferred file will be saved. This is a required option.
ASCII	specifies that the file stored on the PC is stored in ASCII format. It has to be converted to EBCDIC during the transfer to MUSIC. When sending text files, you should always specify the ASCII parameter.
CRLF	specifies whether carriage return/line feed characters are recognized as record separators and deleted before storing the data onto MUSIC.
APPEND	Allows you to attach the PC file to the end of a MUSIC file. For a very large file, you can pre-allocate the file on MUSIC (use (NORLSE,RECFM(VC))), then use the APPEND option.
RECFM	specifies the record format of either V, VC, F, or FC.
LRECL	specifies the logical record length. n can be from 1 to 32xxx.
SPACE	specifies the space needed up to 4000K.
CUT	This option must be specified if your terminal is operating in CUT mode. DFT mode is the default and needs no specification.

RECEIVE : Transferring a file from MUSIC to the PC.

1. Sign on to your MUSIC system from a host session in 3270 mode.
2. Make sure the screen is cleared. You can either press the CLEAR or PA2 key to do this. Failure to do this will cause an error message to appear. Follow instructions at the end to recover from this situation.
3. Press the JUMP key located on the left hand side of the keyboard to go to the PC session. If you have windows defined, the window must be expanded to full size.
4. Issue the following command when the DOS prompt appears:

```
RECEIVE toname  fromname  [ASCII ]  [CRLF ]  [APPEND ]
                   B:fromname
```

where:

toname	is the name of the PC file where the transferred file will be saved. This name can include a drive specification and a path. (Refer to "Specifying the Path to a File" in your IBM Personal Computer DOS manual for further information.) This is a required option.
fromname	is the name of the file being sent from MUSIC. This is a required option. The "B:" is needed for PC/3270 session ID.
ASCII	specifies that the data should be stored in ASCII format on the PC. A translation from EBCDIC to ASCII will be done.
CRLF	specified whether carriage return/line feed characters should be inserted as the last two characters in each line when a MUSIC file is stored on the PC.
APPEND	Allows you to attach a MUSIC file to the end of a PC file.

Notes

1. Failure to have cleared the screen prior to starting the file transfer will cause the following error message to appear on the PC session.

```
TRANS08 Command Transfer Error: File Transfer Cancelled.
```

To recover from this condition you must:

- a. Go to the host session. You should see the message `More...` in the lower right hand corner of the screen.
- b. Press ENTER. The screen will clear and the cursor will move to the lower right hand portion of the screen.
- c. Press the PA1 key THREE times.
- d. Press the ENTER key. The file transfer program (on the host) will be terminated. The PC side of the file transfer program was already cancelled.

You may then restart the file transfer request.

2. Refer to the *IBM 3270 Personal Computer Control Program User's Guide and Reference* (SC23-0102) for more information on using the 3270/PC and file transfer using the 3270/PC.

ASCII Terminals (Connection type 2)

Terminology

MUSIC supports the Teletype Models 33, 35, 38 and 43. (*Teletype* is a trademark of Teletype Corporation, Skokie Illinois.) Other models may also work. Some other manufacturers produce terminals which may work on MUSIC in a similar fashion to those made by Teletype Corporation. The term *TTY* is used as a generic name referring to this general class of ASCII terminals. In many cases the operation of ASCII terminals on MUSIC is independent of its manufacturer.

MUSIC supports the IBM 3101 ASCII display terminal in character mode. This terminal is one of the many that form the TTY class of terminals supported by MUSIC. The IBM publication *IBM 3101 Display Terminal Description* (GA18-2033) contains full details of the keys, switches and operation of this device.

MUSIC supports the IBM 3161, 3163 and 3164 ASCII display terminals in either native or 3101 mode. You can obtain a complete description of the setup, operation, and capabilities of these terminals in the appropriate IBM publications.

Trmcls

The correct specification of the *trmcls* parameter on the /ID command is quite important for the correct operation of some ASCII terminals. Consult the writeup under the /ID command in *Chapter 5 - MUSIC Commands* for more details.

For example, on an IBM 3101, the sign-on command would be "/ID userid;3101".

On an IBM 3161, the sign-on command would be "/ID userid;3161".

Special Keys

Most installations allow users to use the RETURN key as the end of line signal. (If your installation does not, then you must generate this signal by pressing the Q or S key at the same time as you are holding down the CTRL key. You will find it easiest to press the CTRL key first and while keeping it down, press the S or Q key.)

Some models have a separate BACKSPACE key. On others, the backspace function is accomplished by typing the letter "O" at the same time as holding down the SHIFT key. A left pointing arrow (←) or an underscore (_) character will usually print when this is done. The carriage may not actually backspace, but the equivalent function is carried out by the computer.

The BREAK key is used to perform the break function. It can also be used to cancel an entire line, prior to pressing the RETURN key.

The REPT key can be used to ease the typing of the same character many times. Just press the desired key at the same time that you are holding down the REPT key. Release the keys to stop the repeat action.

The upward pointing arrow character is interpreted exactly the same as the vertical bar (|) character on other terminal types.

Terminal Set Up and Connection

Some terminals can operate at different speeds depending on a switch setting. Sometimes the speed switch is marked in characters per second (cps) which is roughly 1/10 the baud rate.

If the terminal has a switch marked full duplex (FDX) or half duplex (HDX), choose half duplex. If the coupler has a half/full duplex switch, choose full duplex.

Establish the connection to the computer. MUSIC may immediately type a message asking you to sign on. If no message appears, press the RETURN key which will then cause the message to be displayed.

You should now sign on without undue delay. Waiting longer than twenty-five seconds to sign on may cause MUSIC to disconnect you.

Special Features and Considerations

TAB characters may be sent to the computer by pressing the letter I key while holding down the CTRL button. The carriage on your terminal may not actually move but the computer will take the proper action.

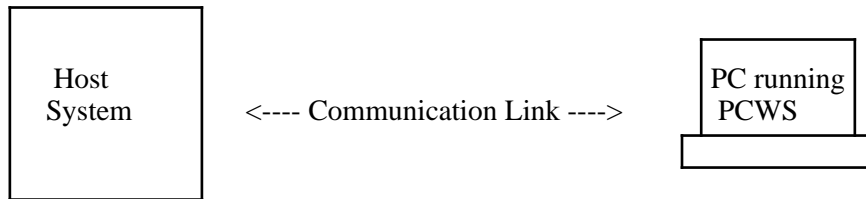
Controlled Scrolling for ASCII Video Terminals

Scrolling is a function of video display terminals by which, as new lines are added at the bottom of the screen, old lines automatically disappear (scroll) off the top. When doing output to a terminal of this type, the situation where the output scrolls off the top of the screen before you have a chance to read it can occur. This problem can be avoided using the `/CTL LINES=nnn` command. (*nnn* specifies the number of lines on the video display screen).

When the screen fills up, MUSIC will ring the bell at the terminal and pause. To continue displaying output, press the RETURN key (or equivalent key). Alternately, your installation may have set up a special terminal class to be used with the type of terminal you are using. (See the `trmcls` parameter of the `/ID` command). In this case when the screen fills up the message `More...` appears in the lower right hand corner of the screen. The RETURN key can be pressed to continue when convenient. On terminals connected over packet switched networks, you may have to press the BREAK key instead of the RETURN key to continue output.

PCWS (Connection type 3)

MUSIC's PCWS (Personal Computer WorkStation) is a Terminal Emulator Program designed to run under the DOS operating system on computers of the IBM PC and PS/2 families. PCWS allows a PC to communicate with the outside world through one of its serial ports.



Starting with PCWS version 2.30, the file PCWS.EXE contains the executable program. (Earlier versions of PCWS consisted of two parts: CLM.COM and TERM.COM.)

For more detailed information, please refer to the *MUSIC/SP Personal Computer Workstation User's Guide*.

PCWS Terminal Types

The PCWS program makes it possible for a PC to emulate one of the following two terminal types:

PCWS Terminal Type

The PCWS terminal type is a special one, designed specifically for communicating over an asynchronous line with the MUSIC operating system. This terminal type has two different modes of operation:

1. PAGE Mode
2. 3270 Mode

The initial mode of the PCWS terminal type is known as the PAGE mode. In this mode, the PC basically emulates a standard line-by-line ASCII terminal. PAGE mode remains the normal mode of operation until a MUSIC full-screen application is started. The PCWS terminal type then automatically switches into 3270 mode. This second mode allows full-screen applications (such as the Editor Mail, etc.) to run on the PC as if it was a real 3270 terminal. Upon terminating the 3270 application, PCWS automatically switches back into PAGE mode.

VT100 Terminal Type

The VT100 terminal type of PCWS allows a PC to emulate the DEC (Digital Equipment Corporation) VT100 terminal. This VT100 emulation capability enables a PC to be connected via a protocol converter.

A protocol converter is a device allowing ASCII terminals to connect to a host computer as if they were IBM 3270 terminals. A PC using the VT100 terminal type of PCWS appears to the host as a regular 3270 terminal. The converter's role is to translate the 3270 data streams it receives from the host into ASCII control sequences the terminal recognizes, and vice versa. As for PCWS, it makes sure the PC processes incoming and outgoing data just as a real VT100 terminal would.

A protocol converter holds a 'Device Definition Table' (DDT) for every ASCII terminal it supports (including VT100). Converters such as the 7171 and the 9370 ASCII Subsystem are shipped with default DDTs for a number of terminals. Each of these supplied DDTs allows a different terminal type to be connected to the converter. During the connection procedure, the protocol converter must find out which DDT to use. To do so, it issues the following prompt:

```
"ENTER TERMINAL TYPE:"
```

A special DDT called 'VT100P' is distributed with MUSIC and should be installed on your site's converter by a system's administrator. The entries of this table provide enhanced key definitions and screen colors.

Other PCWS Features

PCWS was designed specifically to exploit the level of interfacing between MUSIC and the PC. As an example, file transfer is initiated by entering the XTMUS or XTPC command from the MUSIC session. The PCEXEC command of MUSIC executes PC applications as if they were entered at the DOS prompt. Also, many features included in the MUSIC Mail program are facilitated when connected with PCWS.

PCWS for Windows

The original PCWS is designed to run under the DOS operating system. Another program called PCWS for Windows also exists which comes with its own Windows formatted help files.

NET3270 for Personal Computers (Connection type 4)

The Network 3270 Workstation Program provides 3270 terminal emulation to users on local area networks. The NET3270 program also provides multiple session services and file transfer capabilities with the following IBM host operation systems: MUSIC/SP, MVS/TSO, and VM/CMS. NET3270 currently provides 3270 Model 2,3, and 4 terminal emulation which includes extended data stream support, file transfer, HLLAPI (High Level Language Application Program Interface), and support for IBM's GDDM-PCLK Version 1.1.

NET3270 uses either the NETBIOS Session Services transport protocol or Novell's SPX (Sequence Packet Exchange) protocol to provide a communications link between client workstations and NET3270's dedicated communication server.

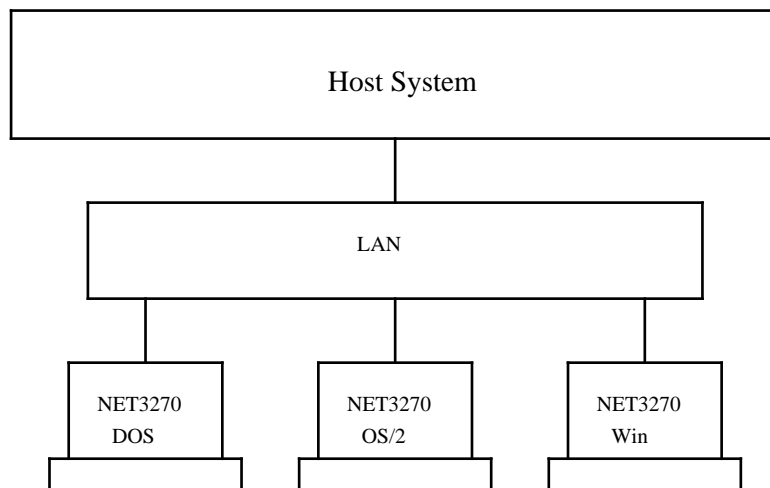


Figure 2.1 - Sample NET3270 Configuration

Starting NET3270

Assuming that NET3270 is installed on your PC and that you are connected to your LAN network, enter "net3270 -r" from your DOS prompt. For example,

```
F:> net3270 -r
```

This will load NET3270 in resident mode (described below). Omit the "-r" if you wish non-resident mode.

NET3270 Features

NET3270 has many features including file transfer and multiple sessions. If you use NET3270 in resident mode you do not need to logout or exit a host application in order to do some work on your PC. You can use the F1 hotkey to toggle back and forth between the host session and the PC session.

The following help text gives you a quick glance at the features of NET3270.

NET3270 QUICKHELP PANEL 1.00			
Alt-A	Create a Session	Esc	Reset
Alt-B	Delete a Session	End	Move to end of field
Alt-C	Clear	Ctrl-End	Erase to end of field
Alt-D	Push to DOS (Non-Res Mode)	Ctrl-Home	Erase Input
Alt-E	Toggle Entry Assist	Tab	Next Field
Alt-F	Change Format	Shift-Tab	Previous Field
Alt-H	Help	Arrow Keys	Cursor Movement
Alt-I	Dump Screen to Disk	Ctrl-W	Delete Word
Alt-J	ALA Print Screen	Ctrl-LfArr	Previous Word
Alt-N	Next Session	Ctrl-RtArr	Next Word
Alt-O	Change NET3270 Features	Ins	Toggle Insert Mode
Alt-P	Previous Session	Del	Delete char at cursor
Alt-Q	Exit NET3270	Ctrl-Enter	Newline
Alt-R	Receive a file	Home	Move cursor home
Alt-S	Send a file	Backspace	Destructive backspace
Alt-T	Sys Request	'-' (Keypad)	PA1
Alt-W	Toggle Word Wrap	'+' (Keypad)	PA2
Alt-F1	Hotkey (Resident Mode)	Return	Enter
Enhanced Keyboard		Normal Keyboard	
Enter (Keypad)	Newline	F1 - F10	PF1 - PF10
F1 - F12	PF1 - PF12	Shift-F1 - Shift-F10	PF11 - PF20
Shift-F1 - Shift-F12	PF13 - PF24	Ctrl-F1 - Ctrl-F4	PF21 - PF24
Press Esc to exit			

Figure 2.2 - NET3270 Help Screen

For further information, refer to the following publications:

NET3270 - The Key to Connectivity - Workstation User's Guide

NET3270 - The Key to Connectivity - Administrator's Guide

NET3270 - The Key to Connectivity - Programmer's Guide

Internet Access (connection type 5)

Using the Internet to Connect to MUSIC

Since the 70's the Internet has evolved to encompass a large number of heterogeneous sites world-wide, and allows you to reach around the world and contact other computer sites. Your computer and the host must be connected via TCP/IP to the Internet in order for you to communicate.

The command you use to access systems with 3270 data streams (like MUSIC and VM) depends on the TCP/IP software that you are using. Two common commands are "TELNET" and "TN3270".

Example of Accessing a MUSIC System

McGill University allows guests to access its CWIS (Campus-Wide Information System) called "infoMcGill". (To use other facilities on MUSIC you need a userid and a password.) InfoMcGill provides the phone directory, library information, positions available, etc. The following example includes McGill University's Internet address.

```
TELNET vm1.mcgill.ca           or           TN3270 vm1.mcgill.ca
```

Once connected, choose "infoMcGill" after the VM logo.

MUSIC's TELNET Command

You can use the TELNET command of MUSIC to connect to any machine on the Internet.

Each site that you visit can be a new system to learn. Follow the instructions carefully. (MUSIC's TELNET has help (F1) when it is first invoked - or type "HELP TELNET" at the *Go prompt.) For more information about TELNET and other Internet commands see the *MUSIC/SP Internet Guide*.

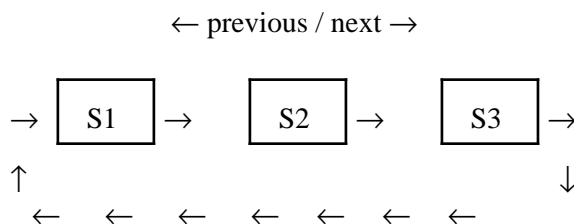
Multi-Session Support

Overview of Multi-Session

Multi-Session Support allows a number of separate sessions to be signed on to MUSIC from one terminal. This support provides the 3270 user with the ability to interrupt one task to perform another. For example, while responding to electronic mail, you realize that your response depends on some information stored in a file. You can suspend the first session and create a new one by pressing a function key. After locating the information you need, press another function key to return to the previous session and resume answering your mail. Another example of using Multi-Session Support involves using a second session while the first session is executing a program.

At any instance only one session is actually in control of the terminal. This is known as the active session.

Function keys are used to create, delete, and switch between sessions. The following diagram represents three sessions.



When you first sign on, one session is allocated. As you create subsequent sessions, they are connected to the original as illustrated above. The functions NEXT (F8) and PREVIOUS (F7) are used to switch active sessions. If S2 is active, then NEXT activates S3, whereas PREVIOUS activates S1. Note that if S3 is the active session NEXT makes S1 active.

The ADD function (F4) creates a new session and connects it after the current active session. The newly created session then becomes the active session. For example, if the ADD function is invoked from S2 then a new session is added between S2 and S3. The number of sessions that a user can create is limited by the number allowed in the user profile and the number of multi-sessions allowed system wide. The ADD function may fail with an error message if either of these limits are exceeded.

The DELETE function (F5) removes the active session from the chain and activates the session preceding it. If S2 is deleted then S1 is the new active session. When a session switch occurs, the system displays the session ID number of the new active session in the message area of the screen. This ID number is the same Terminal number (Tnum) which is displayed by the /STATUS command.

Usage of Multi-Sessions

Multi-Session Support is only available on 3270 terminals and terminals which emulate 3270's. The functions described above are selected by pressing the appropriate program function key. The default definitions of the program function keys are listed below. The /DEFINE command can be used to change these defaults. (See /DEFINE in *Chapter 5. MUSIC Commands* for more information.) To display your current function key definitions enter SHOWPFK in *Go mode.

F4: ADD	A new session is added between the active session and the next one in the chain.
F5: DELETE	The active session is deleted and the previous session becomes the new active session. This is only valid if the active session is in *Go mode (not running any program) and it is not the only session left. The OFF command has the same effect as F5 except it also works for the last session.
F7: PREVIOUS	Make the previous session in the chain the active session.
F8: NEXT	Make the next session in the chain the active session.
F9: PRINT SCREEN	Prints the current screen image on a printer. F9 invokes the REXX procedure called PRTSCR. This procedure copies the current screen image to a file called @PRT and issues the PRINT command. You can use the ROUTE command to change the default printer before pressing F9. Also, you can create your own "PRTSCR" and have the screen image appended to a file instead of printing it.

Programs that use the full screen interface, such as the EDITOR, TODO, and MAIL, have the program function key information passed directly to them, bypassing Multi-Session Support. To inform the system that the program function key is a multi-session request, the user must press the PA2 key before pressing the function key. If the PA2 key is accidentally pressed, the ENTER key can be used to cancel the multi-session request. The following summarizes the sequences required from *full-screen* programs.

PA2, F4 - ADD
PA2, F5 - DELETE
PA2, F7 - PREVIOUS
PA2, F8 - NEXT
PA2, F9 - PRINT SCREEN

The above functions have equivalent MUSIC commands. You can process MUSIC commands by entering them in the command area of a full screen program. The command must be preceded with a slash (/) to inform the current program that you are entering a MUSIC command. For example, /ADD to add a new session. (/DELETE is only valid in *Go mode.)

Notes:

1. Programs continue to run in non-active sessions unless one of the following events occurs: 1) the program requests input from the terminal; 2) the program's output exhausts the supply of output buffers. When one of the above events occurs, the program stops execution until the session is re-activated.
2. From an accounting perspective, using another session is the same as signing on to another terminal. Therefore, users should realize that multiple sessions are charged accordingly.
3. While using the Editor program, it is important to note that changes to the current screen are not recorded when you press PA2. Make sure that you have pressed an action key (i.e. ENTER) before switching sessions.
4. The program named MS can be executed in *Go mode to obtain information about your current sessions. The session ID, userid, and the last command entered are listed for each session starting with the active one.

SESSIONS Command

The SESSIONS command is used to show information about multiple sessions signed on to your userid. The following is an example:

```
----- Current Active Sessions -----
Command ==> _
Point to the session to be Cancelled or Reset and press F5 or F10
      Sess Service Last Conn
  Userid      Id   Units Used Time  Last Cmd Issued, if avail.
-----
? CCGW000      19   92.21  24.9   91.5 print gopher.doc
+ CCGW000      27    IDLE   4.4  106.1 idp
? CCGW000      57    2.70   0.0   25.5 e ug.c2;flag script
* CCGW000      59    1.59   0.0    0.0 sessions

*:Current Session   +:Sessions on another terminal   ?:Hidden sessions
----- THU MAY 13, 1993 09.58.46
F1=Help F3=Exit F5=Cancel Active Pgm F10=Reset Term ENTER=Refresh Lst
```

Figure 2.3 - Screen display for SESSIONS command

Workstation Output Control

Overview of Output Control

The following section contains material that can help speed up the display of information on slow-speed terminals. With the exception of the /CANCEL and /SKIP commands, you may wish to skip this section as the other features described may not pertain to your workstation.

MUSIC has many features that enable the user or the program to control the input and output to the workstation. MUSIC allows you to dynamically skip output, to compress out multiple blanks, or to just display a piece of each output line. The user may also immediately stop the output, enquire how much job time has been used so far, etc.

Dynamic Output Control

MUSIC allows you to interrupt the output by pressing the ATTN or BREAK keys. At this point you can enter any one of a series of commands. Some of these commands can be used to immediately change the format of your output beginning with the next output line. Below is a list of some of the commands you can enter. These commands are further described in *Chapter 5. MUSIC Commands*.

/CANCEL (or /CAN)

This command immediately stops your program and its output and will cause you to return to command (*Go) mode.

/SKIP *n*

This command is used to skip past output lines that you do not want to see displayed at your workstation. The *n* in this command is the number of lines you wish to skip.

/COMP

The COMPRESS command will cause all sequences of multiple blanks characters to display as 1 blank.

/WINDOW

This command can be used to cause only certain sections of each line to be displayed. For example, you could cause MUSIC to display only the first 20 characters of output on each line.

Tabs

MUSIC supports both input and output tabs. The use of output tabs can greatly increase the effective printing speed of your terminal. Output tabs can only be used if the terminal is capable of supporting physical tab settings. These must be set to the same values as specified for the output tabs in the user profile or via the /TABOUT command. The IBM 2741, 3767 and 1050 terminals all have input and output tab capability. The IBM 3270 display terminal does not, though its output speed is so fast that it doesn't need them. Some TTY type of terminals support output tab settings. Input tabs can be useful regardless of the physical tabbing capabilities of the terminal. When inputting on terminals with tab settings, the TAB key is used to skip to the next tab column. In this case the data displayed at your terminal appears as it will when processed by the system. On terminals without physical tabs, a logical tab character can be chosen to provide the tab function, (usually a seldom used character). Pressing this character will not cause the terminal to skip to a specific

column, but when the data is processed, the system will align the data to the correct column. Refer to the writeup on your particular type of terminal for more information. The writeups on the MUSIC commands /TABIN, /TABOUT also contain more information about the use of tabs.

Carriage Control

Each printed line on the terminal starts with a carriage control character. This carriage control character is not printed but is used instead to specify the number of lines to skip before printing the current one. Thus carriage control characters can be used to produce output that is single, double, triple spaced. The convention MUSIC uses follows the standard established for batch jobs. In this way, you can check out a batch job from a terminal and get the same spacing as it would on batch. Some extra carriage control characters have been added to MUSIC to provide more flexible support for some terminals. Refer to the description on carriage controls in *Chapter 4. File System and I/O Interface* for more information.

Direct Terminal Control

MUSIC allows programs to intermix tab skips, line feeds, backspaces, etc., into normal output lines. Additionally, a special carriage control character is assigned so that you can output directly to the terminal bypassing the usual MUSIC translation and output control. (The subroutines NOTRIN and TRIN can be used to control this translation for terminal input.)

These facilities can be useful in some special applications particularly when it is required to control special terminal features such as dynamically setting output tabs, or triggering special graphical features on the terminal. These functions may require different handling depending on the type of terminal being used. Therefore, your program should first verify the terminal type by a call to the MUSIC system subroutine TSUSER to ensure proper output operation of your program.

Workstation Error Messages

*Enter BREAK-time command (or blank line to continue)

The system was expecting one of the break-time commands and the user has entered something else. Typically, this BREAK mode was entered by the user pressing the BREAK or ATTN key. On ASCII (TTY) terminals, it can also be inadvertently entered by the user typing information into the terminal before the system is ready to read it.

*Invalid command

The command just entered is undefined or invalid in the current mode.

*Command not valid at this time

The command just entered is a valid command but cannot be used in the current mode.

*Invalid parameter

A parameter entered on a command is not defined for that command or has a value that is not in the correct range.

*Missing parameter

The command just entered requires a parameter that has not been included.

*Line too long, retransmit

A line longer than 80 characters has been entered, or an undefined tab setting has been used for input, or the total number of characters typed for the line (including backspaces) exceeds the limit of 100. For conversational reads during program execution, the limit is 250 characters including backspaces.

*Invalid character, retransmit

An invalid character has been entered. This can be caused by a terminal transmitting incorrect parity. ASCII (TTY) terminals should be set up for EVEN parity transmission. Other parity bit combinations may also be acceptable.

*Rejected

The user has attempted to cancel a non-cancellable program.

*TRANSMISSION ERROR

An error has occurred on the telecommunications line, for either input or output. If received just after a line has been sent then the line will probably have to be retyped.

*No more sessions chained, Use /OFF

This Multi-Session error message occurs when you try to delete (F5 by default) the last session. There are no other sessions on your terminal at this time.

*Delete is only valid from *Go mode.

This Multi-Session error message occurs when you are trying to delete a session (F5 by default) but you are not in command (*Go) mode.

*Multi-session limit exceeded.

This Multi-Session error message occurs when you attempt to add more sessions than allowed in your user profile. This can also occur when the /DISCON command is used.

*Insufficient storage to create new session.

The system does not currently have enough storage to create the control blocks required for a new session.

*Function key not defined, application continued.

PA2 was pressed in full screen mode entering Multi-Session mode. The function key you pressed was not defined to move to a new session, so the request has been ignored and the original program resumed from where PA2 was pressed.

*There are no sessions chained.

This Multi-Session error message occurs when you have pressed either F7 (previous session) or PF8 (next session) when no other sessions are active.

Chapter 3. Using Batch

Batch Concepts

MUSIC processes jobs from a batch facility at the same time as it is processing jobs from workstations. This batch facility is usually located at the main computer site.

You can submit batch jobs directly from your workstation using the SUBMIT facility described below. Your site may have an alternate method or procedure for submitting MUSIC batch jobs.

Some installations allow MUSIC users to submit batch jobs to other operating systems they may be running and to retrieve output from the job. This is discussed later on in this chapter.

Since many other people may be sharing the same batch facility with you, you might have to wait for some time until your job is run. On the other hand, your MUSIC batch job can be working without you being present to supervise its progress. An installation may wish to give lower priority to jobs submitted to batch, so as to best serve the users who are running jobs from the workstation. Submitting a batch job to run overnight results in better service for all users during the day.

The batch facility features a high-speed line printer that can typically print 132 characters per line at speeds in the range of 1000 lines per minute.

MUSIC batch jobs have access to your files and User Data Set (UDS) files. In addition, batch jobs can use files located on reels of magnetic tape and mountable disk packs.

Since you are not generally present when your batch job is run, it is important to tell MUSIC about any special requests your job might have such as special paper that is to be mounted. You are required to provide maximum limits of how long your job is to run and how much printed output you expect. This information is useful to let the operators know what to expect from your job. This information is also a safeguard for you in case your job goes astray and wants to print thousands of pages of output instead of the ten pages you thought it would.

MUSIC Commands and Statements on Batch

A number of MUSIC commands have meaning only when used from a workstation. Such commands include those to set input and output tabs and to skip output. MUSIC Job Control statements can be stored in a file. You may use all of MUSIC's job control statements from batch.

Job Control Statements

The following is a list of all the job control statements that you can use from batch. Most of these statements are explained in this chapter. For additional information refer to *Chapter 6. MUSIC Job Control Statements*.


```
/COM  
/DATA  
/END  
/ETC  
/FILE  
/ID  
/INFO  
/INCLUDE  
/JOB  
/LOAD  
/OPT  
/PARM  
/PASSWORD  
/PAUSE  
/SYS
```

MUSIC Commands

The MUSIC commands, which must be prefixed with a slash (/) for use with batch jobs, are listed below. Examples of their usage is explained in this chapter. For details about each command refer to *Chapter 5. MUSIC Commands*.

```
/LIBRARY  
/PURGE  
/SAVE
```

Preparing a Batch Job

All MUSIC batch jobs have a /ID as their first statement and a /END as their last statement. A statement containing your special batch password can be located anywhere in between.

Any requests for special paper or other messages to the operator go on a /PAUSE statement which you can put immediately after the /ID statement

A typical batch job set up is shown below:

```
/ID ...      (userid and job limits)  
/PASSWORD=ABC  
/INCLUDE XYZ  
/END
```

This example shows how to run the program XYZ that is stored in a file. The user's batch password in this example is ABC. MUSIC allows part of your program and/or data to be in a file or User Data Set (UDS) file.

Format of the Batch /ID Statement

The first statement of your batch job must be a /ID. It must have information in specific columns as shown in Figure 3.1 below. Your job is automatically canceled if it exceeds any limit given on the /ID statement.

If your userid is not authorized to use the time specified on the /ID, then MUSIC automatically lowers it to

the maximum that you are allowed. Any time limit given on a /SYS statement is ignored when your job is run from batch.

1	5	21	>29	(column number)
/ID	jobname	userid,	tim,pgs,cds	

Figure 3.1 - /ID Statement for Batch

Columns	Contents
1-3	Always the characters /ID.
5-12	An optional name that you wish to give your job. This name has no effect on your program.
21	The userid followed by a blank or comma.
>29	The Time, Pages, and Cards parameters follow the userid and start in column 30 or later. Allow 3 digits for each parameter and separate each with a blank or comma. <i>tim</i> is the maximum job time in units of 60 service units. Thus 002 would mean 120 service units. The leading zeros are required -- blanks are not the same. If column 32 contains the letter "S" then the time limit will be in service units. Thus 20S would mean 20 service units. The word MAX can also be used. <i>pgs</i> is the maximum number of pages of output. A page is typically 60 lines in length. A number of 999 in these columns means unlimited number of pages. Use leading zeros -- for example, type "050" for 50 pages not " 50". <i>cds</i> is the maximum number of punched cards. A number of 999 in these columns means unlimited number of punched cards. Use leading zeros -- for example, type "050" for 50 cards not " 50".

Printer Control

The printed output of your job is controlled primarily by the carriage control characters in the output. These carriage controls can be used to cause the printer to skip to a new page, double space, etc. These carriage controls are discussed in detail in *Chapter 4. File System and I/O Interface*.

MUSIC automatically skips over the first few lines at the top and bottom of a page. This automatic skipping can be avoided for the duration of your job by specifying the option NOSKIP on a /SYS control statement.

Special Operator Message - /PAUSE

A statement of the form /PAUSE text may be placed after the /ID statement to display the message *text* just before the job is run. The operator has to respond to this message before your job will continue. Only one /PAUSE statement is allowed per job. The text of the message can include tape volume names required by your job and any other special requirement that the operator must tend to.

Return Location Messages

You may place one or more /ETC control statements immediately following the /ID or /PAUSE statement. These control statements will be printed on your output immediately following the /ID line. They can be used for special handling messages to the operator, after the job has been processed and printed, such as where your job is to be returned etc. Notice that these statements do not affect the processing of your job and that they are only printed after your job has been run. Requirements that are to affect the job must be specified on the /PAUSE statement.

Purging Files

Files can be purged from batch. However, your installation most likely does not allow this operation. If purging is allowed, then the set-up is as follows:

```
/ID ...                (userid and job limits)
/PASSWORD=abc
/PURGE name
/END
```

abc is the user's batch password, and *name* is the name of the file to be purged. See the PURGE command for more details.

Unit 9 on Batch

If your batch program tries to read from the workstation during execution, MUSIC will automatically interpret the request as a read from the reader instead. This means, for example, that FORTRAN programs that do a READ from unit 9 will be handled exactly as if it were a READ from unit 5.

Submitting Jobs to MUSIC Batch & Other Operating Systems

MUSIC offers the ability to submit files for processing on MUSIC batch, or other batch processors accessible via VM. The facility also exists for sending the printed output from these jobs back to MUSIC for subsequent inspection. The various programs and commands involved in the submission of jobs and the processing of their output are documented together in this chapter of the manual. Since each MUSIC installation can setup the interfaces in different ways, you should consult your installation for details of what is available to you.

SUBMIT Program

The SUBMIT program submits one or more files as a job to MUSIC batch, or to other available batch processors.

By default, SUBMIT generates the appropriate job control statements and submits file(s) to MUSIC batch. Parameters for these job control statements such as CODE, TIME, and PAGES, etc. can be overridden by parameters specified on the SUBMIT command or on a /INFO statement. Jobs can be submitted to other processors by specifying the processor name on a /INFO statement or in the TO parameter on the SUBMIT command. For example, the processor parameter of /INFO is similar to the TO() parameter of the SUBMIT command. Whatever can be specified on the /INFO statement can also be specified as parameters for SUBMIT.

Usage of Submit

When you are in the Command (*Go) mode or in the Editor, you can enter the following to invoke the SUBMIT program:

```
SUBMIT fn1 [fn2] [fn3] . . . [kw1(value1)] [kw2(value2)] . . .
```

The shortest abbreviation for SUBMIT is SUB. The parameters within square brackets are optional.

Parameters

fn1 fn2 fn3 etc.

These specify the names of the files to be submitted. At least one file name has to be specified. Under the Editor, the special name of **CUR* can be used to indicate the current contents of the file being edited. The first file *fn1* should usually contain a /INFO statement (see description later on in this chapter) specifying where the job is to be submitted. If more than one file name is specified, the files are concatenated (joined) in the specified order to form a single job to be submitted.

kw1(value1) kw2(value2) etc.

These are keyword parameters whose values are substituted into the control statements that are submitted with the job. These parameters are dependent on the particular processor to which the job is being submitted, and are usually used to specify items such as time and page limits, passwords, output destinations, etc. The special keyword parameter *TO(processor)*

can be used to specify the processor (system) to which the job is to be submitted. Consult your installation for a list of valid batch processors. If any keyword parameters are specified, they must follow the file name specifications above.

/INFO Job Control Statement

The /INFO statement is used to inform SUBMIT where the job is to be submitted. Keyword parameters, as described under SUBMIT, may also be included on this command. The benefit of the /INFO statement is that the submit destination and the values to be used in the job control statements do not have to be specified each time the SUBMIT program is invoked. If /INFO is used, it must be specified on the first line of the job (i.e., the first line of the first file to be submitted). The syntax of the /INFO statement is the following:

```
/INFO processor [kw1(value1)] [kw2(value2)] . . .
```

The first parameter is the name of the processor (system) to which the job is to be submitted. It could be specified as *MUSIC* to submit the job to MUSIC batch. The keyword parameters *kw1*, *kw2*, etc. are identical to those described for the SUBMIT program. If the parameters do not fit on one /INFO statement, they can be continued on subsequent /INFO statements.

Notes:

1. If there is no /INFO present in the job and the *TO* parameter is not specified when the SUBMIT program is invoked, the job will be sent to MUSIC batch.
2. Any parameters specified on the SUBMIT command override the corresponding ones specified on the /INFO statement.
3. The first /INFO statement must include a processor name.
4. If the same keyword parameter is specified more than once on the /INFO statement(s), the last one will take effect.

Submitting to MUSIC batch

If the processor name of *MUSIC* is specified, or if no name is specified, the job will be submitted to MUSIC batch. The following is a list of keyword parameters that can be used when submitting to MUSIC batch. The minimum abbreviation is given in upper case letters.

<u>Keyword</u>	<u>Description</u>	<u>Default</u>
Userid	1-16 character userid	Sign-on userid
Code	4 character userid *	Sign-on userid
Sub	3 character subcode *	Sign-on subcode
TIme	Time limit in units of 60 service units	10
PAGes	Page limit	500
CArds	Punch card limit	0
CLass	2 character job class (see description below)	AA

FORms	Special forms code	Standard forms
COPIes	No. of copies of printed output	1
Route	Route code for printed output	System Printer or the default specified in user's Profile
PW	MUSIC batch password	Must be specified if other than the default code is used
MSG	Message to operator	Must be specified for jobs requiring special handling or tapes

- * The "Code" and "Sub" keywords are available for compatibility with earlier versions of MUSIC. Users were restricted to 7 characters (4-char code with a 3-char subcode).

The CLASS parameter is used to specify whether the job requires special handling by the operator and when it should be executed. The following is a list of valid class codes:

AA	No special handling. Run as soon as possible. This is the default.
SA	Requires special handling. Run as soon as possible.
TA	Requires tape mount. Run as soon as possible.
AO	No special handling. Run after 6 p.m.
SO	Requires special handling. Run after 6 p.m.
TO	Requires tape mount. Run after 6 p.m.

The ROUTE parameter is used to specify the location for printed output when submitting jobs to MUSIC batch. The following names are valid:

SYSTEM	Send the output to the VM system printer for processing.
MUSIC	Send the output to the MUSIC OUTPUT facility (to be retrieved at your workstation).
rscsname	<i>rscsname</i> is the name of a RSCS printer where the output is sent for processing. For example: R(PRINTER3) means the output is sent to RSCS and queued for printing on linkid PRINTER3.

The name of a MUSIC-controlled ASCII or 3270 printer can also be specified on the SUBMIT command.

Examples of SUBMIT

The first example is shown in full. Only the actual SUBMIT command is given in the rest of the examples. It is assumed that there is no /INFO statement specified in the job for the first four examples.

1. Submit the file PROG1 to MUSIC batch.

```
*Go
submit prog1
103 RECORDS SUBMITTED.
```

```
*End
*Go
```

2. Submit the file PROG1 to MUSIC batch, overriding the default page and time limits.

```
submit prog1 time(10) pages(200)
```

3. The files X.PROG and X.DATA are submitted to MUSIC batch and the printed output is routed back to MUSIC (i.e., the workstation).

```
sub x.prog x.data route(music)
```

4. The tape job in the file ARCHIVE is submitted to MUSIC batch with a message asking the operator to mount the appropriate tape.

```
sub archive cl(ta) msg(mount archive tape 1)
```

5. If the first line in the file PROG is...

```
/INFO MUSIC PAGES(600) TIME(80) ROUTE(MUSIC)
```

then entering...

```
sub prog
```

in *Go mode would submit the file PROG to MUSIC batch to be executed as soon as possible. The limits for the job would be those set from the /INFO statement and the printed output is routed back to MUSIC.

6. If your file does not have a /INFO statement, then to achieve the same results as example 5, enter the following:

```
sub prog TO(MUSIC) PA(600) TI(80) R(MUSIC)
or
sub prog PA(600) TI(80) R(MUSIC)
```

OUTPUT Management Facility

The OUTPUT Management Facility allows you to inspect batch output from a workstation. Output from MUSIC batch jobs is sent to the OUTPUT facility by specifying the ROUTE(MUSIC) parameter of the SUBMIT command. Other operating systems such as VSE, MVS, and CMS can be set up to send printed output to MUSIC in this way. (Consult your installation for details.)

The OUTPUT Facility allows you to look at and work with files of print data in the MUSIC Print Queue. Each entry in the Print Queue represents the output of a previously submitted batch job, or data resulting from a PRINT command, or other print data sent to the Print Queue. Each Print Queue entry has a unique id number (Idnum), an owner, a name (for example, the job name of the batch job), a route (destination) name, and other attributes.

A systems administrator can view all items in the queue (not just their own jobs). Refer to the *MUSIC/SP Administrator's Reference* for more details.

When you start the OUTPUT Facility (by typing the command "output"), the Print Queue entries you own are displayed on the screen. If there are more entries than can fit on the screen, you can use function keys F8 (next page), F7 (previous page), F4 (top), and F5 (bottom) to move around. You can change some of the attributes of an entry by typing over the fields on the screen. You can work with an entry by typing a request code in the "Req" area at the left of the entry; request codes are V (view the data file), B (browse the data file), C (copy the data to another file), P (print), D (delete), etc. The available request codes and PF keys are summarized at the bottom of the screen. You can also enter an OUTPUT or MUSIC command in the command area provided on the screen. You press F3 to leave the OUTPUT Facility.

You own a Print Queue entry if its owner field is the same as your file ownership id (your userid, without the subcode if any).

Entries with route name MUSIC or HOLD are not scheduled to print. They will be held in the queue until you change the route name or delete the entry. Entries with other route names are usually waiting to print, and will soon be printed and removed from the queue. Any entry that has been in the queue for n days or more is automatically deleted; n is set by your installation (typically n is 10 or 20).

Using the OUTPUT Facility

When your workstation is in *Go mode, the OUTPUT facility is invoked by entering "output". The following screen appears.

----- OUTPUT Facility ----- 1 File							
Command ==> _							
Req	Idnum	Owner	Name	Route	Date	#Cop	Forms #Recs

_	1557	BETTY	PRINT	MUSIC	08APR93	1	2181
-----08APR93 08:17--							
Req Codes: V:View B:Browse E:Edit C:Copy P:Print H:Hold D:Delete							
Keys: F1:Help F3:Exit F4:Top F5:Bottom F7:Up F8:Down							
F9:Locate F10:Refresh F11:All F2,F12:Retrieve Cmd							

Figure 3.2 - OUTPUT Facility Screen

OUTPUT Screen Description

The figure above illustrates the screen display for the OUTPUT facility. Each field on the screen is described below. (If your workstation does not support full-screen applications, see the topic "Non-3270 Support (TTY Mode)" later.

Command =>	A command can be entered in this field. There are two types of commands: OUTPUT commands and MUSIC commands. An OUTPUT command is a special command recognized by the OUTPUT Facility, such as LOCATE, SORT, TOP, BOTTOM. Any other command is assumed to be a MUSIC command and is executed as such. You can indicate a MUSIC command by typing a slash (/) before the command name. For example, "/edit myfile". The / is required only if the command name is the same as the name of an OUTPUT command.
Req	Enter a request code in this field, beside the Print Queue entry you want to work with. Request codes are one letter abbreviations for certain operations, such as V (view the data file), B (browse the data file), P (print), D (delete), etc. Available codes are listed at the bottom of the OUTPUT screen. You can enter several request codes, for several entries. For example, to view 3 entries and delete another, enter 3 V's and a D. When you press Enter or a PF key, the requests are done in order, from top to bottom of the screen.
Idnum	This is a unique, fixed identification number assigned to each Print Queue entry. You cannot change an entry's Idnum.
Owner	This is the 1 to 16 character file ownership id of the MUSIC user who owns this entry. The ownership id is the MUSIC userid, but without the "subcode" part (if any). Only a system administrator can change an entry's owner field. For a print file spooled to MUSIC from

some other system, the owner field is the VM distribution (DIST) field.

Name	This is a 1 to 12 character name associated with the Print Queue entry. For the output of a batch job, it is the job name. For an entry generated by the PRINT command, it is PRINT. For an entry generated by a /FILE statement in a job or program, "/FILE ddname PRT(route)", it is the ddname. For a print file spooled to MUSIC from some other system, it is the "CMS filename" field. You can change the name by typing over it on the screen. The change takes effect when you press the next action key (Enter or a PF key other than F3).
Route	Indicates on which printer the file is to print. You can modify this field to have the job print at another printer, provided it has not yet started to print. The route name is 1 to 8 characters. It must be a valid route name defined for your installation. The special names MUSIC and HOLD normally indicate entries that are not scheduled to print at this time. For other names, the entry and print data are automatically deleted once the job is printed on the indicated printer.
Date	The date (ddmonyy) the print file was created. NOTE: After an entry has been in the Print Queue for more than n days, it is automatically deleted (along with its data file). The number n is defined by your installation. Typical values for n are 10 or 20.
#Cop	Specifies the number of copies to be printed (1 or more). You can modify this field by typing over it.
Forms	Specifies the forms number or name (1 to 8 characters), if the output is to print on special forms. A blank in this field indicates standard forms. You can modify this field. (See your installation for names of special forms.)
#Recs	The number of records (lines of output) in the data file.

OUTPUT Request Codes

A list of available request codes is found at the bottom of the OUTPUT screen. These codes can be entered beside jobs in the "Req" field on the screen.

B (Browse)	Enter a "B" beside the print files you wish to browse. This uses the editor (in BROWSE mode) to display the print data file. See also V (view).
C (Copy)	Enter a "C" beside the print file(s) you wish to copy to your own file(s). You will be prompted to enter the name of the file to which the print data is to be copied. If the target file already exists, you are asked whether to replace it; answer Y (yes) to replace the old file, or N (no) to cancel the copy, or A (append) to copy the print file to the end of the existing file. After a copy operation, you can append other print files to the end of the same target file by entering " instead of a file name, when prompted by COPY; this is useful when you want to combine two or more print files into a single file.
D (Delete)	Enter a "D" beside the print files you wish to delete. Both the Print Queue entry and the corresponding data file are deleted. Be careful when using this request code, since a print file CANNOT be retrieved after it is deleted!
E (Edit)	Enter "E" beside the print files you wish to edit. Note that you cannot use the editor to make changes to the print file, since it is not in your library.
H (Hold)	Enter "H" to put print files on hold. The route location changes to HOLD.
P (Print)	Enter "P" to schedule a file for printing. You can change the route name, if you want, by

typing over it on the screen; the P request code uses the new route name if you do so. The P request is handled according to these cases:

1. If the route name is MUSIC or HOLD (or equivalent), the route name is changed to your default route name. If your default route name is a printer name, the file will be scheduled for printing on that printer, and the file will be deleted after it is printed.
2. If the route name is a name such as PC1, indicating printing on your local PC printer or PC network printer, the OUTPUT Facility tries to print the file immediately. The file is not deleted after printing. You should use the D request code to delete it yourself when you no longer need it.
3. Some other route name normally indicates an actual printer. For these, no action is taken, since the file is already scheduled for printing.

V (View) Enter "V" to view print files. This uses the VIEW program to display the print data file. See also B (browse).

OUTPUT Function Keys

The "Keys" section at the bottom of the OUTPUT screen summarizes how the function keys are used in OUTPUT.

F1 - Help	Provides help on how to use the OUTPUT facility.
F2 - Retrieve	Same as F12 (see below). Retrieves a previous command.
F3 - Exit	Exits from the OUTPUT Facility. PA1 has the same effect.
F4 - Top	Moves to the top (beginning) of the list of Print Queue entries.
F5 - Bottom	Moves to the bottom (end) of the list of Print Queue entries.
F7 - Up	Displays the previous screen of Print Queue entries.
F8 - Down	Displays the next screen of Print Queue entries.
F9 - Locate	Locates the next occurrence of the string specified by the previous LOCATE command. For example, if you previously entered the command "locate fred" in the command area, pressing F9 locates the next occurrence of "fred".
F10 - Refresh	Refreshes the screen with the latest contents of the Print Queue. When OUTPUT starts, it reads the Print Queue and remembers the entries it found. Those entries are displayed, until you press F10 to tell OUTPUT to read the queue again. Refresh may show new entries recently added to the queue, or drop entries that have recently been deleted.
F11 - All	(This key is ignored unless you are a system administrator with the LSCAN privilege.) It displays all entries in the Print Queue, for all users. It is equivalent to the command SCAN FLIP. To revert to the normal display, press F11 again.
F12 - Retrieve	Displays the previous command entered in the command area each time this key is pressed. Up to 10 previous commands can be recalled. When you press F12, any old text in the command area is replaced, without being executed. After retrieving a command, you can modify if you want, then press Enter to execute it, or Clear to clear it from the command area, or F12 to retrieve the previous command.

OUTPUT Commands

The following special OUTPUT commands can be entered in the command area of the OUTPUT screen. If a command you enter is not recognized as one of these, it is executed as a MUSIC command. You can type a slash (/) before a command to indicate that it is to be treated as a MUSIC command, not an OUTPUT command. You can use the Clear key or F12 to move the cursor to the command area.

The minimum abbreviation is shown below each command name.

BOTTOM

B	Displays the last full page (screen) of Print Queue entries.																						
LOCATE string L	Locates the next Print Queue entry that contains the specified character string. The string may appear within any of the displayed fields: Idnum, owner, name, route, date, etc. For example, the command "locate fred" would find an entry with owner FREDDY, or with name JOBFRED, etc. If you do not specify a string, the string from the previous LOCATE is used (this is equivalent to pressing F9), and the next occurrence is located.																						
SORT field order SO	Sorts the displayed entries in ascending (A) or descending (D) alphabetical order by a specified field. The first parameter, field, is one of the following (the first 3 characters of the field name may be used as an abbreviation): <table> <tr><td>IDNum</td><td>Sort by Idnum</td></tr> <tr><td>OWNer</td><td>Sort by owner</td></tr> <tr><td>NAME</td><td>Sort by name</td></tr> <tr><td>ROUte</td><td>Sort by route name</td></tr> <tr><td>DATE</td><td>Sort by date (chronological order)</td></tr> <tr><td>COPIes</td><td>Sort by number of copies (numerical)</td></tr> <tr><td>#COPIes</td><td>Sort by number of copies (numerical)</td></tr> <tr><td>FORms</td><td>Sort by forms name</td></tr> <tr><td>RECOrdS</td><td>Sort by number of records (numerical)</td></tr> <tr><td>#RECOrdS</td><td>Sort by number of records (numerical)</td></tr> <tr><td>OFF</td><td>No sorting (equivalent to sort by Idnum)</td></tr> </table> The second parameter, order, is A or D. If omitted, A is assumed. If you want to sort by a field (say ROUTE), and sort by a second field (say NAME) within that, use two consecutive SORT commands: <pre>sort route sort name</pre>	IDNum	Sort by Idnum	OWNer	Sort by owner	NAME	Sort by name	ROUte	Sort by route name	DATE	Sort by date (chronological order)	COPIes	Sort by number of copies (numerical)	#COPIes	Sort by number of copies (numerical)	FORms	Sort by forms name	RECOrdS	Sort by number of records (numerical)	#RECOrdS	Sort by number of records (numerical)	OFF	No sorting (equivalent to sort by Idnum)
IDNum	Sort by Idnum																						
OWNer	Sort by owner																						
NAME	Sort by name																						
ROUte	Sort by route name																						
DATE	Sort by date (chronological order)																						
COPIes	Sort by number of copies (numerical)																						
#COPIes	Sort by number of copies (numerical)																						
FORms	Sort by forms name																						
RECOrdS	Sort by number of records (numerical)																						
#RECOrdS	Sort by number of records (numerical)																						
OFF	No sorting (equivalent to sort by Idnum)																						
TOP T	Displays the first page (screen) of the Print Queue entries.																						

Order of Operations on the Screen

It is possible to combine several operations at the terminal before pressing an action key. Operations are done by the program in the following order:

1. Print queue entry changes (route name, number of copies, etc.) and request code action, from top to bottom of screen. For a given entry, request code action is done after the entry changes.
2. Command from command area, if any.
3. PF key, if any.

When Clear, F3 (Exit), or PA1 is pressed, any changes to the screen since the last action key are ignored.

Non-3270 Support (TTY Mode)

When the OUTPUT Facility is used on a terminal that does not support 3270-type full-screen applications, a different user interface, called TTY mode (or non-3270 mode), is used. It is a line-at-a-time interface, with commands used for all input from the user. Additional OUTPUT commands are available, to provide the functions done by screen input and PF keys in 3270 mode.

For more information enter the MUSIC command: "HELP OUTPUTTTY".

Printing Files

Files can be printed on any of the line printers defined in the system by using the PRINT command. The PRINT program can be invoked in *Go mode or when you are in the Editor.

The PRINT command schedules the printing of a file to a specified printer. The file name must be the first parameter. The other parameters are optional and may appear in any order. Carriage control is added to skip to a new page every sixty lines unless CC is specified or the file's record length is 121 or 133.

The PRINT command does not send your file for execution. It prints the contents of the file. The Editor also has an similar PRINT command.

Syntax

```
PRINT filename [ROUTE(printername)] [FORMS(x)] [COPIES(n)] [CC  ]
                R                F                C                [NOCC]
                [PAGELEN(m)]
                P
```

If you are in the Editor, the shortest abbreviation for PRINT is PRI. There is no abbreviation in *Go mode. The parameters within square brackets are optional and may appear in any order, separated by a blank.

Parameters

filename The name of the file to be printed. Under the Editor, the special name **CUR* indicates the current contents of the file being edited, and the special name . (period) indicates marked lines.

printername The name of the printer where the file is to be printed. This may be an actual printer name or a printer location. It is 1 to 8 characters long. Some documentation refers to this as a "route name" or "routing name". The names are assigned by your system administrator.

If you do not specify a printer name, a default name is used. If you have used the command "ROUTE printername" previously in this MUSIC session, that name is the default. Otherwise, the name defined by ROUTE(name) in your User Profile (the PROFILE command) is used, if any. Otherwise, a default name based on your workstation location may be used. If none of the above cases apply, the name SYSTEM is used.

The following names are valid:

SYSTEM Sends the output to the standard system printer.

MUSIC Sends the output to the MUSIC Output Queue (the OUTPUT Facility).

DUMMY Discards the output. Nothing is printed.

rscsname The name of an RSCS printer. For example: R(PRINTER3) means the

output is sent to RSCS and queued for printing on linkid PRINTER3.

prtname	The name of a MUSIC-controlled ASCII or 3270 printer as defined by your installation. Consult your installation for a list of valid names.
PC1	A printer name such as PC1 may be defined by your installation. It prints the file on your PC printer (using DOS device LPT1), provided your PC is connected to MUSIC via NET3270 or PCWS. If the connection does not support PC printing, the data is sent to the MUSIC Output Queue (the OUTPUT Facility). Similarly, PC2 uses device LPT2 and PC3 uses device LPT3. When printing to a PC printer, some PRINT parameters such as COPIES, FORMS and PAGELEN may be ignored.
n	The number of copies that should be printed. The default is 1 copy.
x	A one to 8 character string indicating which forms are to be used when printing the file.
m	The number of lines per page. The default is 60 lines. This parameter is used only when the NOCC parameter is in effect.
CC	If specified, it indicates that the file to be printed already contains a carriage control character on the first character of each line in the file. The PRINT command attempts to honour these control characters.
NOCC	Indicates that the file does not contain printer control characters. The file is printed single spaced, with a skip to a new page after each <i>m</i> lines of output. NOCC is the default, except when the record length of the file is 121 or 133, in which case CC is assumed.

Example:

```
*Go
print file1
*In Progress
15 records scheduled to print, route SYSTEM
*End
*Go
```

Checking the Status of Printers

The PQ utility program can be used to find out what is queued to print on the various printers. It can be invoked from *Go mode by entering PQ. It should be noted that if the printer in question is under the control of another operating system, the fact that MUSIC has transmitted the print file to that system and removed it from the print queue does not necessarily mean that the print file has actually been printed. The OUTPUT command is useful in displaying the status of the individual print files.

Chapter 4. File System and I/O Interface

File Systems

MUSIC supports two file systems -- Save Library files and the User Data Set (UDS) files. Most users use only the Save Library file system. Unless otherwise specified, reference to files on MUSIC/SP refer to Save Library files.

Files are easy to create, edit, and manage. They can contain programs, text and data of any kind and sequential, direct and indexed access is supported. Programs can dynamically open and close files. The contents of files can be easily exchanged with other host systems and personal computers.

UDS files use the IBM standard VTOC format established around 1965. It does not offer the same performance, data compression and usability benefits found with the regular MUSIC files. It is mainly used in MUSIC to handle some temporary scratch space and to allow concurrent data sharing with other operating systems in some circumstances. Most MUSIC users will never have to know anything about UDS files. UDS files are created and accessed through the /FILE statement. See *Chapter 5. MUSIC Commands* for more details about the /FILE command that applies to UDS files.

Save Library Files

The Save Library file system (disk) is the main file system on MUSIC. Each file in the Save Library is owned by a particular user. File ownership is based on the ownership id which is the sign-on userid excluding any subcode. If two users share the same userid (different subcodes) then they also share the same files.

You normally have unrestricted access to the files you own. You can share files with other users if the appropriate access controls are specified.

Files can be created using the editor, using /FILE statements, or dynamically by application programs. A file name is assigned to the file when it is created and used to reference the file later on. To edit a file, the file name is specified as a parameter on the edit command. To run a program stored in a file, the file name itself is entered as a command. The LIBRARY and FLIB commands can list the names of all the files that you own. It will also optionally give details as to the size of your files, when they were created, and various other information. See *Chapter 5. MUSIC Commands* for more details about the LIBRARY and FLIB commands and other commands that apply to files.

File Storage Technique

The Save Library is located on disk. Though physically consisting of multiple data sets, the user need never be concerned about where it is located. All users get their Save Library space from a common pool. When a file is purged, the space is automatically released and is made available for other users. The next user of the space you once occupied will be unable to look at what you had once stored there. Information in files is normally stored in compressed format in order to reduce disk space and disk read/write time. This compression is done automatically by the system and need not concern you.

File Names

The user specifies a *file name* when a file is created. The file can then be accessed in future by merely giving the file name.

MUSIC allows the user great flexibility in the choice of names for files. Each file name can be up to 17 characters in length. File names cannot include blank characters. Each character of the name can be any letter (A-Z) or any number (0-9) or any of these special characters:

~ ! @ # \$ % & _ + .

The exception to the above rule is that the first character of the name cannot be any of the following:

~ ! % & _ + .

and should not be the @ symbol.

Notes:

1. Files starting with @ are reserved for possible use by system programs and utilities. Some special rules apply to how they are handled so avoid their use. For example, the name @LIB is used by the LIBRARY command.
2. UDS file names cannot use the following special characters:

~ ! % & _ +

Examples of valid file names are: PROG, PROG.V1, RM.LETTER.05DEC92, REPORT_1, 51980, THISISALONGNAME.

Users who will be saving lots of files may wish to establish some naming convention to help identify the contents of each file. One suggested naming convention is to use names such as PROG.S to contain the source for the program PROG and PROG.OBJ to contain the object module for PROG. If a load module file is needed, it could be named PROG.LMOD. (The TAG command of the Editor can be used to store a descriptive phrase with each file to further help in identifying its contents.)

Hierarchical Tree Structured File Naming

MUSIC supports the popular hierarchical tree structured file naming convention. It is used in the same way as the one used on personal computers running the DOS system. It is also compatible with the one used with the AIX and UNIX systems.

The tree structured method is particularly useful when you have many files. The following describes the benefits and the basic outline of how to use this naming convention. The use of this method is optional on MUSIC. If you do not use this method, then you are using what is called the "flat file system" naming convention.

Suppose you are a university professor. You teach several courses and you work on several research projects. You want to keep your files grouped by subject. You could establish a naming convention that your file names starting with the characters C100 referred to course C100, the ones starting with C200 to another course and VR would mean your files about your research project in virtual reality.

You could use names like C100.NOTES and C200.NOTES to keep the files separate between your two courses. Alternately, you could use the features of the MUSIC system to keep the files separate. You would do this by making "directories" for each course. One could be called C100 and the other C200. You make the directory for the C100 course by issuing the "MD C100" command. The other is made by using the "MD C200" command.

When you want to work on course C100, you change the directory by issuing the "CD C100" command. MUSIC will respond with the prompt of "*Go \C100>" indicating that you are in that directory. Now you

when you issue the "EDIT NOTES" command, the system knows you want the one associated with the C100 course. By default when you create new files, they will go into your "current directory" (C100 in this case.) You can issue the "DIR" command to list all the files stored under this directory.

Note that you can have a file called NOTES under different directories. They will all be different. In that way, your notes for course C100 are kept separate from the C200 course. The system does this by prefixing the name of your current directory in front of the files you use. So for example, editing the file NOTES under the C100 directory causes the system to work on the file "\C100\NOTES". You could edit the same file by typing in the command "EDIT \C100\NOTES" but that involves typing more characters.

Suppose you are editing the file NOTES as described above and you wanted to merge in the notes from the C200 course. This can be done by issuing the command "MERGE \C200\NOTES".

You can even define directories within directories. That allows you to further organize your files. Suppose you setup a directory called "VR" to contain the files to do with your virtual reality project. You want to have separate directories for your research results, the papers that you are writing about your research, and the letters asking for additional grants to support your project. Now you have a collection of directories which are linked in a hierarchical manner. You can use the MUSIC TREE command to help you visualize the organization of these directories. See the following diagram. This structure will resemble a tree with each directory being another branch in the tree. You travel around the tree by issuing the CD command to change the current directory. You can issue the command "CD \" to get back to the "root" of the tree where you have no current directory prefix being added to your files.

```
----- Change/Remove/Make Directory -----  
  
Place cursor on new directory and press ENTER to change directory  
-----  
userid:\ ----+---- C100  
          |  
          +--- C200  
          |  
          +--- VR --+--- LETTERS  
                   |  
                   +--- PAPERS  
                   |  
                   +--- RESULTS  
  
F1=Help F3=End F7=Up F8=Down F10=<- F11=-> F4=MD F6=RD F9=DIR F12=FLIB
```

Figure 4.1 - Screen display for TREE command

Suppose you are in directory "\VR\LETTERS". You could have got there by issuing the "CD VR" followed by the "CD LETTERS". Now suppose you want to go work on your files in the C100 directory. You do that by issuing the "CD \C100" command. If you had issued the command "CD C100" the system would have thought you meant go to directory "\VR\LETTERS\C100" The "\" character at the beginning of the name is a signal to the system to go to the root before searching for the requested directory.

To remove any directory use the RD command.

There is a limit of 50 characters to hold the file name plus the directory prefix on MUSIC. That means that

you can have several levels of directories within directories but not an infinite number of them!

You can learn more about the CD, MD, DIR, RD and TREE commands by consulting the writeup on each command in *Chapter 5. MUSIC Commands*.

As mentioned under the previous topic, you should not use file names starting with "@". Those names bypass the use of the current directory prefix.

Common and User Indexes

The system maintains an index (list) of all files that the user creates in the Save Library. A different index is maintained for each user. This index may be listed by using the LIBRARY command.

In addition the system maintains a common index to facilitate the sharing of files among different users. Some users are allowed to place a file name in this common index so long as no other user has previously used that name (see "Access Control" below).

Note: File names for MUSIC programs and commands are in the common index. If you have a "private" file with the same name as a "common index" file, then you will access your file only. Use the RENAME command to rename your file if necessary.

Search Order (File Names)

When a user refers to a file, the system will first check the user's index to see if the file exists. If not, then the common index will be searched. This feature allows users to use other people's public files as simply as his or her own.

If the user has used the CD command to specify a current directory then that directory is searched first. If the file is not found there then the user's root directory is searched. If still not found then the common index is searched. (On MUSIC, this searching is a very fast operation. It does not read through lists of files like many other systems must do.)

If a file name has the share (SHR) attribute, it is possible to prefix a file name with an ownership id (userid excluding any subcode) as in the example OWNERID:PROG to directly refer to another user's file. Using a file name such as *COM:PROG will cause a search of the common index only.

The user's current directory will not be searched if the file name is prefixed with the character "\", or includes the owner's userid as in the example "OWNERID:PROG", or starts with the special character "@".

Access Control (File Attributes)

Access to any file is controlled by control information stored with each file. This control information is set depending on the attributes you give when the file is created. The most commonly used ones are described below.

Saving a file with the *private* attribute means that only the owner of the file can access it. This is automatically the default. No entry in the common index is made in this case.

Saving a file with the *public* attribute means that other users can read the file. It may only be modified or deleted by its owner. This allows programs and data to be shared among many users while preventing unwarranted destruction of information. The name of the file is placed in the common index.

Saving a file with the *share* attribute means that other users can read the file but they must prefix the file

name with the owner's userid as in the example USERID:PROG. The file may only be modified by its owner. The name of the file is **not** placed in the common index.

The *execute-only* attribute can be used to further restrict access. It means that programs can be executed but not accessed in any other way. For example, a file with the public and execute-only attributes can be shared among many users while preventing inspection and copying it.

Some users may be restricted from saving files with public and share attributes. This is usually done to discourage file sharing between users and to avoid the creation of nuisance files in the common index.

Record Formats

The record format is used to specify the format and characteristics of the records in the file. MUSIC supports the following types of record format for files:

- F Fixed Format. All records of the file have the same record size.
- FC Fixed Compressed. All records of the file appear to have the same record size. Internally, the file is compressed, with four or more consecutively repeated characters reduced to two bytes, to save disk space and disk read/write time. This type of record format is the most commonly used one.
- V Variable Length. All records of the file may have different length.
- VC Variable Compressed. All records of the file may have different length. Internally, the file is compressed, with four or more consecutively repeated characters reduced to two bytes, to save disk space and disk read/write time.
- U Undefined Format. This is used by some system utility programs that do their own record handling.

Record Size

The maximum record size of any file is 32760 bytes long.

File Quotas

Your installation may have set a limit to the total amount of space your userid can use in the Save Library.

The biggest file you can create is 56 million bytes. Your installation may have set a lower limit than this for your userid.

The LIBRARY command can be used to show the amount of space used by each file and the total used by all your files.

Note: In addition to MUSIC commands there are several utility programs available to help you with files. See *Chapter 10. Utilities* for more information.

Unit Numbers

The MUSIC system can read and write from many different places. For example, it can read from a disk file and write on a high-speed batch printer. Unit numbers are used in many cases to simplify the specification of what devices are to be used.

FORTTRAN programs and utilities can be made to read and write from different units simply by using different unit numbers. For example, unit number 6 can be used to direct output to the workstation. The MUSIC default unit number convention is detailed below. The user may alter these specifications through the use of /FILE statements.

(COBOL, PL/I, VS ASSEMBLER and similar programs use *data definition names* (ddnames) instead of unit numbers. These processors have default ddnames. Refer to the write ups on these compilers for information about these defaults.)

- 6 Unit 6 is the printer. If the program is being run from a workstation, then specifying this number will cause the output to be directed back to your workstation. If the program is being run from batch, then this unit number directs the output to the batch printer. The maximum record size is 250 with the first character being a special carriage control character. Refer to the following discussion on carriage controls for further details.
- 7 Unit 7 is the card punch (it is rarely used anymore). This unit number is mainly used if your job is being run from batch. The maximum record size is 80.
- 5 Unit 5 is used to read files. The /INCLUDE statement can be used to tell MUSIC to read from the named file. Your job may use many /INCLUDE statements if you wish. (/INCLUDE statements may occur anywhere in your file.) The maximum record size is 80.
- 9 Unit 9 is used to read input conversationally from your workstation. Your job will temporarily pause waiting for you to respond to the read. From batch, a read on this unit number is taken as if it were one on unit 5. The maximum record size is 250. Users on 3270 terminals can only enter a maximum of 80 characters.
- 10 Unit 10 can be used to temporarily hold output from a program until it can be saved when the job is over. You can save the output from this unit with the MUSIC command "/SAVE name,SV" when your workstation is next in *Go status. The maximum record size is 80. See the discussion later on the Holding File for more details about this file.

Carriage Control Characters

Overview

The first character of each line printed via default unit number 6 is considered to be a carriage control character. This carriage control character is not printed but is used instead to specify the number of lines to skip before printing the current one. Thus carriage control characters can be used to produce output that is single, double, triple spaced, etc. MUSIC has extended the list of valid control characters to provide more flexible support for some workstations.

If an invalid carriage control is detected, the line will be effectively shifted over by 1 and a blank carriage control will be assumed. This automatic handling of incorrect carriage controls is performed merely as a convenience to the user. As other carriage controls may be defined from time to time, the user should not rely on this automatic correction procedure.

Supported Carriage Controls

The following lists the supported carriage control characters and their effect. Some of them have particular meaning only on certain types of workstations. Some use a non-printable *hexadecimal* carriage control character to avoid possible accidental conflicts with incorrect carriage controls.

blank	A blank for a carriage control results in normal spacing.
0 zero	Double spacing. This control will print a blank line first before printing the current line.
- minus	Triple spacing. This control prints two blank lines before printing this one.
1 one	Skip to new page. This control is supported on batch. Many workstations do not support an equivalent feature, in which case, this control will be taken as equivalent to a double space request. IBM 1050 terminals may be able to perform this operation. IBM 3270 terminals will cause the current line to be displayed at the top of the output area. The 3270 will go to MORE status first if the output area contains some output from previous writes. Certain TTY video terminals will work similarly to the IBM 3270 terminals.
+ plus	Overprint. Supported only at batch. All terminal types will handle this as if it were a blank carriage control.
hex '70'	Erase Screen. Causes an immediate erase screen operation on 3270 terminals and certain TTY video terminals. On all other workstations, it will be taken as a blank carriage control. This carriage control may not clear the screen until some further output is generated by the program. Simply writing a line that is hexadecimal 70 followed by a hexadecimal 00 will be sufficient to clear the screen if no other program output is generated. This carriage control should not be used on batch.
hex '71'	Write to the message area (for 3270-type workstations only). The message area is the last line of the screen. The output line will be indented 9 spaces and the maximum length is 60 characters. An output line longer than 60 characters will be truncated.
hex '72'	Compress multiple blanks to a single blank, and also remove any blanks at the beginning of the line. The resulting line is displayed using single spacing. This control character is

effective only for workstations, and should not be used on batch.

- hex '79' Write to the input area (for 3270-type workstations only). The first 80 characters following the control character are written to the 3270 input line. This carriage control should not be used on batch.
- hex '41' Direct output control. Causes the characters following this special carriage control to be transmitted to the workstation without the usual checking, translation and idle character generation. Since translation is not done, the output sequence must be in the required code for the specific type of workstation involved. This is used as part of the direct workstation control feature of MUSIC. Sending incorrect sequences may result in messages such as *TRANSMISSION ERROR or may even cause your line to be disconnected. This carriage control should not be used on batch.
- hex '62' Sounds the alarm (beep) on a 3270-type workstation.

Holding File

WARNING: The holding file is now of limited use and is provided mainly for compatibility for users accustomed to older versions of the MUSIC system. A /FILE can be used to specify that the unit 10 output is to be directly written to a file.

Users have access to a special holding file during the execution of their programs. This file can be used to hold the object modules produced from the compiler when the DECK option is used from a job running at a workstation. A user's program can also write output to this file via a write on unit 10.

The holding file is actually a file that is stored under your userid. It has the special name of @HOLD. (If your userid includes a subcode then the name is @HOLD.sub where sub is your 1 - 8 character subcode.)

After the job is finished, you can then save the contents of this file. You do this by using a command like "/SV name" or "/SAVE name,SV".

Alternately you can refer to it by the special name of /HOLD. This allows you to type in commands such as "RENAME /HOLD myobj".

A program can automatically cause an appropriate save command to be issued when the job ends. The system subroutine SAVREQ can be used for this purpose. Even if such an automatic save operation should fail, you still have the chance of entering a correct save command. (Use of the system routine NXTPGM does not give you this second chance.)

Spooled Conversational Reads

Introduction

MUSIC's normal conversation read facility (default I/O unit number 9), offers the user the maximum degree of interaction between the user and the running program. When a read 9 is performed, the program is stopped until the read is complete. At this time, the job resumes execution and can process the information just entered at the workstation. The program can decide whether to perform another read operation, or issue a message, etc.

Sometimes this degree of interaction is not required. Suppose for example, a program reads 10 lines in succession from the workstation and then processes this information. Using normal conversational reads will mean that the program will be interrupted 10 times -- once for each read. Only one interruption is needed. MUSIC addresses this requirement by providing the *Spooled Conversational Read* facility. This facility can be used to read a number of lines at a time while only requiring one interruption.

Considerable performance improvement can be achieved using this spooled facility since your job will require fewer time-slices. Additional flexibility is provided by the optional ability to process /INCLUDE control lines that may be entered at this time. A maximum number of lines to read in this mode can be given, or the user can end the set of reads by entering a blank line.

The MUSIC Editor is an example of a program which uses this facility. It uses spooled conversational reads when it is in INPUT mode for older models of terminals (non-full screen).

Usage

The spooled conversational reads are always done via the MUSIC I/O unit number 5 not 9. The maximum record length is 80. This facility is not available when the job is run at batch. Also, use of spooled conversational reads cancels the effect of any previous call to subroutines NXTPGM, SAVREQ, and SIGNOF.

The program can detect the end of the set of lines read from the workstation by checking for the blank line or an end-of-file condition. The blank line will be present except when the reads were stopped by the program-specified line count, or it exceeded the maximum that could be stored in the system spool file. (This system spool file can hold at least 300 lines, so this should not pose much of a limit. You can, of course, re-issue the spooled read request to read in another set of 300 lines in this case.)

This spooled read facility is requested by the program via a call to the system subroutine SYSINR before each set of lines is read. This SYSINR subroutine is also used for other dynamic read facilities. The following gives the usual calling sequence. For further details consult *Chapter 9. System Subroutines* of this manual.

```
CALL SYSINR( '/TRMIN  ',ln,lln,nc,k)
```

ln This number is normally given as 1.

lln This number is the maximum number of lines to be read. The number -1 specifies no fixed limit specification.

- nc This number is normally 1.
- k This number is usually given as 2 or 3. The value 3 means that /INCLUDE statements will be processed if found. The setting of 2 or 3 will give an end-of-file indication at the end of the set of lines read. Add 4 to get control of errors.

User Data Sets (UDS)

User data sets are used to conveniently hold large files consisting of data, programs or load modules. A single file is limited only by the size of an entire disk volume. This type of file can be accessed by FORTRAN Direct Access.

UDS files can be allocated as temporary or permanent. Permanent files can be removed (scratched) when you are finished with them. Temporary files are used for the duration of one job only.

UDS files can be manipulated by using the Editor, a user's application program, or by utilities and subsystems provided with MUSIC.

Batch users can access UDS files stored on magnetic tape or those located on mountable disk packs.

You need a /FILE statement in any program that refers to a UDS file. Each record of a UDS file can be up to 512 characters in length. This *record size* is specified on your /FILE statement. Some processor or application programs may impose a lower limit to this record size.

File Storage Technique

UDS files are stored on disk just like direct access data sets are stored by the OS and MVS operating systems. They have standard VTOC entries and data is not compressed on them by the system. The minimum file size is one track long.

UDS File Names

Each UDS file is referred to by a character data set name (dsname) of up to 22 characters in length. In addition a disk pack volume name must be given. There are two types of dsnames:

- Type 1 A dot (.) is used to separate the ownership id from the name. The dsname can be up to 22 characters long. For example: "GEORGEW.TEMP1".
- Type 2 When there is no dot (.) in the dsname then the first 4 characters of the dsname must match your userid. The maximum length of the dsname is 8 characters. For example: "CCGWTMP1".
Type 2 is available for compatibility with older versions of MUSIC.

UDS Access Control

Access to a UDS is governed by the use of an indicator character included in the dsname. This character is in the 1st position after the dot (type 1) or in the 5th position (type 2). The indicator may be one of the following:

- \$ private - only the owner is allowed to read or write to it (Example: GEORGEW.\$TEMP or CCGW\$TMP).
- # public - any user can write to it.
- x read only - x represents any other letter or number and it means that any user can read but only the owner can write or delete.

Refer to the /FILE statement write up for more details.

UDS Quotas

The installation may limit the maximum UDS file size that you can create.

Note: There are several utility programs available to help you with User Data Sets. See *Chapter 10. Utilities* for more information.

Storing Data on a UDS File

Data can be copied from a file(s) to a UDS file using the program UTIL as described in the topic "UTIL" in *Chapter 10 - Utilities*.

If desired, both object modules and data may be stored on the same UDS file just so long as there is a /DATA control statement separating the object modules from the data. The program can read from the file using the I/O unit number given on the /FILE statement.

UDS files can be accessed using sequential or direct access input/output (I/O) statements. A UDS file created by one program using sequential I/O techniques can be used by another using direct access. A direct access data set written by one job, however, should not be read sequentially from another job as there will be no end-of-file marker after the last data record.

If you are careful, you can even use both the sequential and direct access techniques in one program. You cannot use both methods at the same time. Just do a rewind operation before starting to use the file in a direct access manner. Call the system subroutine CLOSDA before switching to a sequential access manner.

Disk Block Utilization

MUSIC disk UDS files always use 512-byte blocks. MUSIC will often read and write them in groups, thus handling them as if they were 1024 through 10240 bytes in length. This multiple block read/write feature of MUSIC is performed automatically.

When a UDS file is created, you specify its record size. MUSIC uses this number together with the number of records you wanted in order to calculate how many 512-byte blocks to set aside for your file. For example, if your record size was 128 bytes, then it knows it can fit 4 per block. A record size of 80 bytes means that 6 will fit per block leaving 32 bytes unused.

Buffer Allocation for UDS Files

Though buffer allocation for UDS files is automatic in MUSIC, the following explanation may be of assistance.

Each UDS file used in a job is allocated at least two buffers each. MUSIC may allocate 4 or even 6 buffers to a single UDS file. These additional buffers allocated automatically by MUSIC can considerably increase the efficiency of your program in some cases. It may be to the user's advantage to ensure that adequate space is left for buffers.

Buffers for UDS files on disk are always 512 bytes each for a total of 1024 bytes per pair of buffers. Each buffer for a UDS file on tape is equal to the blocksize (BLK) given on the /FILE statement defining it.

FORTTRAN direct access data sets cannot use more than one pair of buffers per data set, although more may

be allocated for them.

Available Buffer Space - UDS

The main storage available for buffers for UDS files (including files on magnetic tape) that may be used depends on which processor is being used and the size of the user region. The following descriptions are arranged by the name given on the /LOAD statement. If the specific compiler is not listed, then it probably is the same as /LOAD ASM.

ASM	Buffer space is whatever is required for tape and UDS buffers (2 buffers each), and is a minimum of 4096 bytes. A large enough user region must be used so that sufficient space remains for the program and GETMAIN area. Assuming a large enough user region, tape block sizes may be up to 32760.
EXEC	Buffer space available is approximately the user region size, less 15K, less the length of the load module. The handling of buffers for tape files is described under FORTG1 processor given below.
FORTG1	<p>Buffer space available during the compile and loading of the program is 4096 bytes.</p> <p>If the user specifies a /FILE statement for a magnetic tape file, then the system will attempt to allocate buffers for it when the job starts. If there is insufficient space at that time, it will try again once your program is loaded. The job will be terminated at this time if insufficient space remains.</p> <p>Buffer space available during the execution phase is approximately the user region size, less 13K, less the size of the user program (not counting IBCOM). Buffers are reallocated just before the execution is to begin to take advantage of any extra space that may be available.</p>
LKED	<p>Buffer space during the Linkage Editor phase is the same as that described for /LOAD ASM. The Linkage Editor requires one UDS file or one file to hold the load module it produces. Unit number 4 is also used as a temporary UDS file during the link edit process.</p> <p>Buffer space during the execution phase is described above under the EXEC processor.</p>
LOADER	The buffer space for the LOADER is identical to that of FORTG1.
PLC	Buffer space available is 4096 bytes. The user may specify up to 3 UDS files.
PLI	<p>Same as /LOAD ASM.</p> <p>The user can use 3 UDS files. A UDS file on unit 4 contains the compiler modules and run-time transient library.</p>
PLILG	<p>Same as /LOAD ASM.</p> <p>The user can use 3 UDS files. A UDS file on unit 4 contains the PL/I transient library modules, which are loaded dynamically during execution.</p>
VS BASIC	Buffer space available is 4096 bytes. This space may be increased by using the BUF=n parameter on the first /OPT statement (refer to <i>Chapter 8. Processors</i> for the section on VS BASIC).
XMON	Same as /LOAD ASM.

XMPLI Same as /LOAD ASM.

The user can use 3 UDS files. A UDS file on unit 4 contains the PL/I transient library.

MUSIC/SP Virtual Storage Access Method (VSAM)

Introduction to VSAM

MUSIC/SP VSAM (Virtual Storage Access Method) is an implementation of the data organization method described in the IBM publication *OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide*, GC26-3838. That publication describes the features and usage of VSAM from the point of view of an assembler programmer. Most of the assembler interface documented there is supported on MUSIC/SP for programs running in OS simulation mode. Some major features not supported by MUSIC/SP are: control interval access, spanned records, data set passwords.

The VSAM assembler interface is also used by higher level languages which allow access to VSAM files. In this way, MUSIC/SP programs written in languages such as PL/I, VS Cobol, and VS Fortran, in addition to VS Assembler, can use VSAM.

MUSIC/SP supports the three types of VSAM files: KSDS (key-sequenced data set), ESDS (entry-sequenced data set), and RRDS (relative record data set). The data records, which make up the *data component*, of a VSAM file are stored in a single file or UDS (user data set) file. In the case of a KSDS, the *index component* is stored in a second file. Alternate indexes (which are themselves KSDSs) can be built over a KSDS or ESDS. The Access Method Services (AMS) utility program is used to create and initialize VSAM files on MUSIC/SP.

The most significant feature of VSAM is its ability to access data records by means of a *key*, which is a fixed-length field at a fixed displacement in each data record. Each key, along with a pointer to the data record which contains the key, is also stored in a multi-level *index* (the primary index or an alternate index), which can be quickly searched for a given key. Instead of specifying a record number or an address on disk, the user program specifies the key when retrieving or updating a record. For example, a student number key could be used to identify a student record in a file of marks, or a part number could identify a part record within an inventory file. Records can be added, changed and deleted; VSAM automatically updates the index as required. It is also possible to access the data records sequentially, in order of increasing or decreasing keys.

VSAM References

OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide, GC26-3838.

Introduction to IBM Direct-Access Storage Devices and Organization Methods (student text), GC20-1649. Refer to the chapter on VSAM.

OS/VS2 Access Method Services, GC26-3841. On MUSIC/SP, a subset of Access Method Services is provided by the utility program AMS, which is described in *Chapter 10. Utilities*.

VSAM Abbreviations

ACB	Access control block. An OS control block which is specified in order to open a VSAM file (logically connect a program to a VSAM file). The ACB is analogous to the DCB (data control block) for non-VSAM files.
-----	--

AMS	Access Method Services utility program.
CA	Control area. A logical group of control intervals.
CI	Control interval. A fixed-length area on disk, where VSAM stores data records. The CI corresponds to a logical block for non-VSAM files.
ESDS	Entry-sequenced data set. One of the 3 types of VSAM files.
KSDS	Key-sequenced data set. One of the 3 types of VSAM files.
RBA	Relative byte address. The displacement (expressed as a fullword binary integer) of a data record or control interval from the beginning of the data set. The VSAM RBA of the first control interval is 0.
RPL	Request parameter list. An OS control block which defines an input/output operation to be performed on a VSAM file.
RRDS	Relative record data set. One of the 3 types of VSAM files.
RRN	Relative record number. The record number (1, 2, 3,...) of a data record (slot) in an RRDS.

VSAM Data Storage and Organization

A KSDS consists of a data component and an index component. An ESDS or RRDS consists of a data component only (ignoring alternate indexes for the moment). The data component and index component each occupy one file on MUSIC. It is possible for the data component, but not the index component, to be a UDS (user data set) file instead of a regular file.

For example, a KSDS cluster could comprise the 2 files SAMPLE1.DAT (the data component) and SAMPLE1.IDX (the index component). Any file names can be used, but a naming convention such as this is recommended. The first 512-byte block of SAMPLE1.DAT contains the file name of the index, along with other control information. The user refers to the KSDS cluster by giving the file name of the data component, SAMPLE1.DAT, usually on a /FILE statement.

MUSIC/SP does not have a separate VSAM catalog. Attributes and control information for a VSAM file are stored in the first 512-byte block of the data component and index component files. User data starts in the second 512-byte block. For example, if the control interval size is 4096, which is eight 512-byte blocks, then the first data CI occupies blocks 2 through 9 of the file and is at VSAM RBA 0. The 2nd CI is blocks 10 through 17, at VSAM RBA 4096, etc. Control information is in block 1.

The VSAM RBA is different from the MUSIC/SP file system (MFIO) RBA. They are related by the formula: MFIO RBA = VSAM RBA + 1024.

An *alternate index* is a special type of KSDS which is associated with a KSDS or ESDS base cluster. It allows access to data records via a key field different from the primary key. Block 1 of the alternate index data component contains the file name of the data component of the base cluster. There can be several alternate indexes built over a KSDS or ESDS. The combination of an alternate index and a base cluster is called a *path*. Each path is given its own name and is represented by a 1-block file of that name. The path file contains the file name of the alternate index data component. So, when a base KSDS is accessed via a path, there are 5 MUSIC/SP files involved: the path file, the data and index components of the alternate index, and the data and index components of the base KSDS. The name of the path file is specified on the /FILE statement.

A MUSIC file which is part of a VSAM file normally has bit X'20' on in the first byte of the 4-byte access control field.

VSAM control intervals on MUSIC/SP have the same format as on OS, except that spanned records are not allowed.

VSAM indexes on MUSIC/SP have the same structure as on OS, and index records have the same basic format, except for the following differences. The first index record (record 0) starts at the 2nd 512-byte block of the file. Block 1 contains control information. There is only one *section* per index record. In the sequence set (lowest level), vertical pointers are always 3-byte CI numbers, relative to the beginning of the data component, which starts with CI 0. Sequence set records do not contain pointers to free CIs, since new CIs are always added after existing CIs in the data component. Sequence set records are chained backward as well as forward. A chain of free index records is maintained.

Features Not Supported by VSAM

Compared with OS VSAM, the following restrictions apply to MUSIC/SP VSAM.

- Records may not span control intervals. This means that the longest record that can be stored is CILEN-7 bytes, where CILEN is the data component CI length. CILEN can be up to 32768 if necessary, but is fixed for a given file. This also restricts the number of pointers that can be stored in an alternate index record, for an alternate index with the NONUNIQUEKEY option. LRECL for an alternate index should be defined as CILEN-7 to allow for records as large as possible.
- Control interval access (MACRF=CNV and OPTCD=CNV) is not supported.
- Passwords for VSAM files are not supported.
- Reusable data sets (VSAM work files) are not supported. This is the REUSE option of the AMS DEFINE command.
- These performance options are not supported: IMBED, KEYRANGES, REPLICATE.
- The maximum number of alternate indexes in the upgrade set is 10. Note that, as on OS, performance is slow when updating a base cluster with several alternate indexes in its upgrade set, so the number of alternate indexes should be kept to a minimum.
- A data component is limited in size to the maximum file, which is usually 57000K, unless the data component is a UDS. However, an alternate index cannot be built over a UDS base.
- There are limitations on how VSAM files can be shared by multiple users and/or programs. In general, concurrent updating by multiple users or programs requires special handling. See the section on file sharing below.
- The OPEN/CLOSE message area is not supported. If a message area (MAREA and MLEN parameters) is defined for an ACB, it is ignored by MUSIC/SP.
- MUSIC/SP VSAM is available in OS simulation mode only. It cannot be used from Fortran G1 programs or other programs running in MUSIC (non-OS) mode.
- Only the following types of exit routines are supported: EODAD (end of data set), LERAD (logical error), SYNAD (physical error). An exit routine must not be specified as the member name of a module to be loaded dynamically by VSAM (the L option of EXLST).

- The following macros for advanced applications are not supported: BLDVRP, DLVRP, GETIX, MRKBFR, PUTIX, SCHBFR, SHOWCAT, VERIFY, WRTBFR.
- The following fields are not supported in SHOWCB and TESTCB macros for an ACB: AVSPAC, BFRFND, BUFNO, BUFRDS, ENDRBA, FS, HALCRBA, NCIS, NDEL, NEXCP, NEXT, NINSR, NIXL, NLOGR, NRETR, NSSS, NUIW, NUPDR, STMT, UIW.

VSAM Usage Notes

A VSAM file to be used by a program must first be created and initialized by the DEFINE command of the AMS utility. At that time, you specify file characteristics such as name, type of file, amount of disk space required, data and index CI length, length (1 to 255) and position of the key field, and maximum logical record length. Refer to the description of the AMS utility for details.

For most applications, recommended index and data CI lengths are 512 and 4096, respectively. If the key length is 234 or more, the index CI length must be at least 1024.

In order to use the file in a program, a /FILE statement of the following form is required:

```
/FILE ddname NAME(filename) disp
```

ddname	is the 1- to 8-character data definition name used in the program. This name is specified in the ACB when the file is opened.
filename	is the name of the data component file. In the case of accessing a base cluster via an alternate index, it is the name of the path file.
disp	disp is the disposition. For read-only access, it should be SHR. For read/write access, it should be OLD NORLSE. The NORLSE option tells MUSIC/SP not to release unused space when the job ends. NORLSE can be omitted if you do not expect to make major additions to the file after this job. To allow concurrent access by multiple users or programs, some of which may be updating the file, the disposition should be WSHR NORLSE for read/write, otherwise SHR.

Main Storage Requirements - VSAM

VSAM services during a user program are provided by the re-entrant modules VSAM (about 32K) and CBMANIP (about 14K). They are loaded from the system Load Library or Link Pack Area the first time the job requires them. They occupy main storage in the user region, unless your system administrator has placed them into the Link Pack Area. Module VSAM is always required for VSAM. CBMANIP is only required if any of the control block manipulation macros (GENCB, MODCB, SHOWCB, and TESTCB) are used. VSAM programs written in high-level languages such as PL/I require CBMANIP.

In addition to storage for modules, VSAM requires a work area of about 10K in the user region.

VSAM requires data buffers and index buffers (in the case of a KSDS) for each base cluster and alternate index cluster. They are allocated in the user region when the VSAM file is opened.

The length of each data buffer is the data CI length. There is 1 data buffer for each concurrent string (for most applications the number of concurrent strings is 1). An additional 1 or 2 work data buffers are required if a KSDS or alternate index base is being updated. These additional buffers are allocated automatically and should not be included in the number of buffers specified in the ACB or in the AMS DEFINE command.

The length of each index buffer is the index CI length, normally 512. There are no index buffers for an ESDS or RRDS. For a KSDS, there are at least 4 index buffers. Specifying more index buffers will usually increase performance, especially if the index is large and access is not sequential.

Each alternate index cluster is a KSDS, and therefore has its own set of data and index buffers.

Each open cluster requires approximately 2K additional storage for miscellaneous control blocks and work areas.

VSAM File Sharing

A job that updates a VSAM file while other jobs are using the file must specify WSHR, rather than OLD, on the /FILE statement. Using OLD prevents all other jobs from accessing the file while the current job is running.

When WSHR is specified, it is not possible to add space to a file. Only the currently allocated space can be used.

When one job is updating a VSAM file while other jobs are accessing it for read-only, read integrity for the other jobs is the responsibility of those jobs. See the discussion of cross-region share option 2 in the VSAM Programmer's Guide. Also, it is up to the updating job to use ENQ or a similar facility to prevent other jobs from updating the file at the same time.

If multiple jobs are allowed to update a file concurrently, read and write integrity is the user's responsibility. See the discussion of cross-region share options 3 and 4 in the VSAM Programmer's Guide. In general, multiple concurrent updating is not recommended unless ENQ is used to serialize all requests and the ENDREQ macro (or CLOSE TYPE=T) is used to release buffers and positioning. MUSIC/SP VSAM does not prevent concurrent updaters from ignoring these requirements and damaging data.

The CLOSE TYPE=T request is useful when sharing files. It forces modified data buffers and block 1 buffers to disk, releases positioning, invalidates the main storage copies of index buffers, and rereads block 1 information from disk, but the data set remains open for further processing. It is logically similar to a CLOSE followed by an OPEN, but is more efficient. The ENDREQ macro does not invalidate index buffers or update block 1 information.

MUSIC/SP does not support the *shared resources* feature of OS VSAM. This refers to the sharing of control blocks and buffers among several VSAM data sets open at the same time.

VSAM Error Codes

VSAM open, close, and input/output requests return an error code number if the request does not complete successfully. In some cases, the error codes for MUSIC/SP differ slightly from those for OS VSAM. The error codes for MUSIC/SP are listed below.

For open and close, the error code is returned in the byte at displacement 49 in the ACB. For I/O requests, the error code is in the 4th byte of the feedback word in the RPL. The feedback word is at displacement 12 in the RPL; the error code byte is at displacement 15. The register 15 (R15) return code for an I/O request is in the 2nd byte of the feedback word. An error code of 0 indicates successful completion.

OPEN Error Codes When R15 Is Not 0

Note: Always test for an error condition after an OPEN, since attempting to use an ACB which was not opened successfully usually causes a branch to location 0 (a program interrupt).

- 4 (Warning only.) The data set is already open.
- 128 The ddname is undefined, or an indicated file cannot be found.
- 144 An I/O error occurred while reading or writing the control information in block 1 of a VSAM file.
- 148 Unable to open a file, for a reason other than the conditions in OPEN error codes 128, 152, 168, and 172.
- 152 Requested access to a file is not allowed (MFIO error code 43).
- 160 Inconsistent or invalid control information in block 1.
- 168 File is in use (MFIO error code 33).
- 172 Too many open files (MFIO error code 20).
- 188 The block 1 identifier is incorrect, or the file is not a VSAM file, or a path file does not point to an alternate index, or an item in the upgrade set is not an alternate index.

CLOSE Error Codes When R15 Is Not 0

- 4 The data set is already closed.
- 144 An I/O error occurred while writing block 1 control information to disk, or (for CLOSE TYPE=T) while reading block 1.
- 148 Unable to close a MUSIC/SP file.
- 184 An I/O error occurred while VSAM was completing outstanding I/O requests.

I/O Request Error Codes When R15=8 (Logical Error)

(For error codes not shown here, refer to the VSAM Programmer's Guide.)

- 4 End of data set encountered. Either no EODAD routine is provided, or it returned to VSAM and the processing program issued another GET.
- 8 You attempted to store a record with a duplicate key, or there is a duplicate record for an alternate index with the unique key option.
- 16 Record not found.
- 32 You specified an RBA that is not the address of a data record.
- 44 For a GET with OPTCD=MVE, the receiving area is not large enough for the data record. The data is truncated and the GET completed normally except for this. For a GET with OPTCD=LOC, the receiving area is less than 4 bytes long.

- 64 The request cannot be started because the number of active requests would exceed the maximum number of strings.
- 68 Attempt to PUT or ERASE on a read-only file. The file may be read-only because SHR or WSHR was used on the /FILE statement, or because the ACB specified input processing only.
- 72 Keyed request on an ESDS is not allowed.
- 76 Insert by RBA is not allowed on a KSDS.
- 80 ERASE on an ESDS is not allowed.
- 84 OPTCD=LOC is not allowed for a PUT request or in an RPL in a chain of RPLs.
- 88 Required positioning does not exist, or you attempted an illegal switch between forward and backward processing. For example, you issued a sequential GET without having caused VSAM to be positioned for it.
- 92 PUT for update or ERASE is not preceded by a GET for update.
- 96 An attempt to change the prime key or key of reference when updating a record.
- 100 An attempt to change the length of a record when making an addressed (by RBA) update.
- 104 Invalid or conflicting RPL options.
- 108 Invalid record length for a PUT. It is negative, or 0, or larger than the maximum, or is too small to contain the entire key field, or is not equal to the slot size for an RRDS.
- 112 Key length is 0 or too large.
- 144 Invalid pointer (no associated base record) in an alternate index.
- 148 Unable to add a pointer to an alternate index record during upgrade processing, because the resulting length of the alternate index record would exceed the maximum record length or CILEN-7. The original PUT request is not done.
- 192 Invalid relative record number (RRN) for a request on an RRDS.
- 196 Addressed request is not allowed on an RRDS.
- 200 Addressed access is not allowed through a path.
- 204 PUT insert request is not allowed during backward processing.

I/O Request Error Codes When R15=12 (Physical Error)

- 4 Read error in a data component.
- 8 Read error in an index component.
- 16 Write error in a data component.
- 20 Write error in an index component.

I/O Error Message Area Contents

When a physical error occurs (return code R15=12 for an I/O request) and the RPL specifies a message area which is 128 bytes or longer, MUSIC/SP VSAM fills in the message area as described below. The first 12 bytes are the same as in OS VSAM. The remaining bytes are in printable format, and are different on MUSIC/SP and OS. Only the first 128 bytes are filled in. No bytes are filled in if the message area length is less than 128.

Bytes 0-1: Binary value 128 (message length).

Bytes 2-3: Unused (0).

Bytes 4-5: Binary value 124.

Bytes 6-7: Unused (0).

Bytes 8-11: I/O buffer address.

Bytes 12-127: Printable text in the format:

```
VSAM dddddddd t-cccc I/O ERROR nnn: xx BLK=mmmmmmmm
LEN=11111 BUF=aaaaaa yyyyy...
```

Bytes 17-24: (ddddddd) The file's ddname.

Byte 26: (t) B for base cluster, A for alternate index cluster, or U for alternate index cluster being accessed during upgrade processing.

Bytes 28-32: (cccc) DATA for an data component or INDEX for an index component.

Bytes 44-46: (nnn) MUSIC/SP file system (MFIO) error code.

Bytes 49-50: (xx) RD for a read operation or WR for a write operation.

Bytes 56-61: (mmmmmm) The starting 512-byte block number where the error occurred.

Bytes 67-71: (lllll) The length used in the MFIO I/O request.

Bytes 77-82: (aaaaaa) The buffer address used in the MFIO I/O request.

Bytes 84-127: (yyyyy...) The first 44 characters of the error description text corresponding to the MFIO error code. This describes the reason for the error.

VSAM Abend Codes

When VSAM detects an unexpected error which it cannot recover from, it issues an abend (abnormal end) message on logical unit 6 (i.e. the workstation or whatever is defined as /FILE 6). The message contains an abend code number, which indicates the reason for the abend.

The format of the abend message is:

```
*VSAM ABEND nnn AT aaaaaa INFO = xxxxxxxx xxxxxxxx xxxxxxxx
```

nnn is the abend code. *aaaaaa* is the address from which the abend routine was invoked. *xxxxx....* gives 3

words of additional information, in hexadecimal. The meaning of the INFO words depends on the abend code.

Following the abend message, VSAM enters a conversational dump routine or dumps the user region (for a job running on batch). Then the job is terminated.

For a complete listing of the abend codes, list or edit file VSAM.ABENDS.

Some notable abend codes are:

- 5 Not enough index buffers.
- 6 Unexpected error while retrieving a data record from a control interval. This error usually indicates invalid control information in a data CI.
- 7 Attempt to access a spanned record. Spanned records are not supported.
- 10 Attempt to use control interval access. CI access is not supported.
- 14 Illegal change of request options in an RPL chain.
- 17 An invalid index record has been encountered. The first word of INFO has the index record number in hexadecimal.
- 19 Invalid record in the data component of an alternate index cluster.
- 22 Chained RPLs are not allowed with a PUT/update or ERASE request.
- 26 For a PUT request, the address of the logical record in the RPL (the RPL AREA field) is 0. I.e. no area address has been provided.

Tracing and Debugging Facilities

Two options on the /JOB statement (or on the member name statement for /LOAD XMON) can be used to trace and debug the internal workings of the VSAM processing modules. In some cases these options are useful in determining why a user program is failing.

The IOTRACE option causes a trace message to be written to logical unit 6 for each entry to, or exit from, a VSAM processing routine. Since many routine calls are made for each user request, this option may generate a very large amount of output.

If the CDUMP option is used with IOTRACE, a conversational dump routine is entered after each VSAM trace message. Commands, read conversationally from logical unit 9, are used to inspect and modify main storage and registers, to skip a specified number of subsequent trace items, and to turn VSAM tracing on or off. Enter a blank line to resume execution. To suppress all further VSAM tracing, enter the command TRACE END and then a blank line (non-VSAM tracing will continue).

VSAM Miscellaneous Notes

- For a GET request, if the address in the RPL of the receiving area for the logical record (the RPL AREA field) is zero, then no data is moved but the GET is completed normally and no error indication is given. This applies to both move and locate mode.

- It is possible to open a VSAM file dynamically. Normally a ddname is specified in the ACB when a file is opened. In that case the file must be defined by a /FILE statement. A special form of ddname is used to open a VSAM file which is not defined on a /FILE statement. This feature is available only on MUSIC/SP. The 8-byte ddname field is, in hexadecimal, 0Fxx0000 00aaaaaa, where 0F indicates this special type of OPEN, xx contains option bits, and aaaaaa is the address of a 22-character area which contains the full file name of the data component or path file. Bits in the option byte are: X'80' on means the cluster is to be opened read-only (equivalent to SHR on /FILE); X'40' bit is used internally by VSAM OPEN; X'20' on means that the cluster is to be opened for shared read/write (WSHR on /FILE). Option byte X'00' opens the cluster for non-shared read/write (OLD on /FILE).
- The OS RDJFCB macro (read Job File Control Block, SVC 64) can be used before OPEN to test whether a ddname represents a VSAM or non-VSAM file. This is useful to know, since opening a VSAM file requires an ACB while opening a non-VSAM file requires a DCB. On MUSIC/SP, the byte at displacement 99 (JFCDSRG2) in the JFCB for a VSAM file has bit X'08' on. After the RDJFCB, do a normal OPEN, not OPEN TYPE=J. Refer to RDJFCB in *OS/VS2 System Programming Library: Data Management*, GC26-3830.

PL/I Support

The environment option REUSE is not supported by MUSIC/SP.

The environment option BUFND(n) should not be used, since extra data component buffers will not improve performance. The BUFSP(n) option is not supported. Use BUFNI(n) to specify the number of index buffers if you wish to use more than the standard number.

The VSAM Compatibility Interface, which allows a VSAM file to be accessed by a PL/I program originally written to access an ISAM file, is not supported.

Sample Program for VSAM

The sample assembler program presented here uses the macros described in the VSAM Programmer's Guide to copy records from a sequential file, MYDATA, to a VSAM KSDS file, VSAMFILE1.DAT. The records are added as new records.

The following job creates and initializes the data and index components of the KSDS using the AMS utility. The key field occupies the first 8 bytes of each record. The initial space allocation for the data component is 20K. The data control interval size is 4096. The name of the index component file is VSAMFILE1.IDX. The index control interval size is 512 by default.

```
/INCLUDE AMS
DEFINE CLUSTER(NAME(VSAMFILE1.DAT) INDEXED SPACE(20) -
    CISZ(4096) KEYS(8 0))
```

The sample program is listed below. It is contained in file VSAM.SAMPLE.

```
/FILE DDNAME1 NAME(MYDATA) SHR
/FILE DDNAME2 NAME(VSAMFILE1.DAT) OLD NORLSE
/LOAD ASM
*
* THIS SAMPLE PROGRAM READS DATA RECORDS FROM A SEQUENTIAL
* FILE ON DDNAME "DDNAME1" AND WRITES THEM TO A VSAM KSDS
* ON DDNAME "DDNAME2". IT IS ASSUMED THAT THE VSAM FILE
```

```

* HAS ALREADY BEEN CREATED AND INITIALIZED.
* THE INPUT RECORDS ARE ASSUMED TO BE OF LENGTH 80.
*
* AN OPEN ERROR CAUSES AN INVALID OP-CODE WITH R7=X'EE1' AND
* R0=OPEN ERROR CODE.
*
* AN ERROR WRITING TO THE VSAM FILE CAUSES AN INVALID OP-CODE
* WITH R7=X'EE2', R0=ERROR CODE FROM PUT, R15=RETURN CODE FROM PUT.
*
SAMPLE    CSECT
          REGS      ,          THIS MACRO DEFINES REGISTERS
          STM      R14,R12,12(R13)  SAVE REGISTERS
          LR       R12,R15          SET UP A BASE REGISTER
          USING    SAMPLE,R12
          LA       R14,SAVEAREA     SET UP A SAVE AREA
          ST       R13,4(0,R14)
          ST       R14,8(0,R13)
          LR       R13,R14
* OPEN THE TWO FILES
          OPEN     (MYDCB,INPUT)     OPEN SEQUENTIAL FILE
          OPEN     (MYACB)           OPEN VSAM FILE, USING AN ACB
* GET ERROR CODE FIELD FROM ACB
          SHOWCB   ACB=MYACB,AREA=OPENERRC,LENGTH=4,FIELDS=(ERROR)
          L        R0,OPENERRC
          LTR      R0,R0             TEST FOR VSAM OPEN ERROR
          BZ       OPENOK           BRANCH IF NO ERROR
* VSAM OPEN ERROR
          LA       R7,X'EE1'        INDICATE INTENTIONAL P.I.
          DC       H'0'             STOP JOB BY INVALID OP-CODE
OPENOK    DS      0H
* READ LOOP: READ NEXT RECORD INTO "MYAREA"
READLOOP  GET     MYDCB,MYAREA      READ RECORD USING QSAM
          PUT      RPL=MYRPL        WRITE RECORD TO VSAM FILE
          LTR      R15,R15          TEST FOR WRITE ERROR
          BZ       READLOOP         BRANCH IF NO ERROR
* VSAM ERROR: GET ERROR CODE FROM RPL
          LR       R5,R15           SAVE R15 RETURN CODE
          SHOWCB   RPL=MYRPL,AREA=REQERR,LENGTH=4,FIELDS=(FDBK)
          L        R0,REQERR        GET ERROR CODE IN R0
          LR       R15,R5           RESTORE R15 RETURN CODE
          LA       R7,X'EE2'        INDICATE INTENTIONAL P.I.
          DC       H'0'             STOP JOB BY INVALID OP-CODE
* COME HERE WHEN END-OF-FILE ON INPUT: CLOSE THE FILES
EOF       CLOSE   (MYDCB)
          CLOSE   (MYACB)
* WRITE MESSAGE TO TERMINAL
          WTO     'VSAM SAMPLE PROGRAM ENDED NORMALLY'
* RETURN TO SYSTEM
          L        R13,4(0,R13)     RESTORE SAVE AREA POINTER
          LM       R14,R12,12(R13)  RESTORE REGISTERS
          BR       R14             RETURN
* STORAGE AREAS
SAVEAREA  DS      18F             STANDARD SAVE AREA
OPENERRC  DS      F               RECEIVES OPEN ERROR CODE
REQERR    DS      F               RECEIVES ERROR CODE FROM RPL
MYAREA    DS      CL80            LOGICAL RECORD BUFFER

```

```

* DCB FOR SEQUENTIAL INPUT FILE
MYDCB      DCB      DDNAME=DDNAME1,DSORG=PS,MACRF=GM,LRECL=80,EODAD=EOF
* ACB FOR OPENING VSAM FILE
MYACB      ACB      DDNAME=DDNAME2,MACRF=(KEY,SEQ,OUT)
* RPL FOR REQUESTS ON THE VSAM FILE
MYRPL      RPL      ACB=MYACB,AREA=MYAREA,AREALEN=80,RECL=80,          X
                  OPTCD=(KEY,SEQ,NUP)
END

```

