MSM-Server for UNIX

System Manager's Guide

Version 4.4

March 1998

Micronetics Design Corporation

Disclaimer

Micronetics Design Corporation makes no representations or warranties with respect to this manual. Further, Micronetics Design Corporation reserves the right to make changes in the specification of the product described within this manual and without obligation of Micronetics Design Corporation to notify any person of such revision or changes.

The software described in this document is furnished under a license by Micronetics Design Corporation and may only be used or copied in accordance with the terms of such license.

Micronetics Design Corporation
1375 Piccard Drive
Rockville, Maryland 20850

Telephone: 301-258-2605
Fax: 301-840-8943
E-Mail: info@micronetics.com
WWW: www.micronetics.com

Contents

MSM-NET Services 163

MSM-LAT Terminal Services 201

Preface

Acknowledgment

Micronetics Standard M (MSM) is an implementation, with extensions, of the ANSI Standard Specification (X11.1-1995) for the Massachusetts General Hospital Utility Multi Programming System (MUMPS). MUMPS was developed by the Laboratory of Computer Science at Massachusetts General Hospital under grant number HS00240 from the National Center for Health Services Research and Development. MUMPS was a trademark of the Massachusetts General Hospital.

Documentation Conventions

The following documentation conventions are used in this manual.

Convention	Description
RETURN	The carriage return key (normally labeled RETURN, ENTER, and so on).
CTRL+X	The CTRL key pressed at the same time as the X key, where X is any valid key used in combination with the control key.
<ERROR>	An MSM error message.
'val'	In Help messages, 'val' is used to indicate that the user can enter the indicated value. The value is entered without the quotes.
>	The MSM Programmer prompt.
...	The series of items repeats a user-specified number of times.
. . .	Shows a break in a list where consecutive lines have been omitted.
Bold	Items in a dialogue are shown in **bold** to indicate a user response.

Getting Started

Overview

This chapter describes the distribution package contents, hardware and software requirements for MSM-UNIX, and storage requirements.

System Requirements

In order to install MSM properly, certain minimum system requirements must be met. The following sections outline these requirements and the minimum amounts necessary to satisfy them. The following table summarizes the minimum requirements. Memory requirements vary based on the make and model of the computer hardware, size of the UNIX operating system, and application size.

Minimum System Requirements

System Requirements	All Systems
Minimum system memory	32 MB
Floppy diskette (3.5-inch)	Optional
Minimum hard disk space	16 MB
Standard serial/parallel ports	Optional
Cartridge or 9-track tape drive	Optional
LAN adapter (Ethernet, and so on)	Optional
LAT terminal servers	Optional

Hardware Requirements

MSM-UNIX runs on UNIX, AIX, DG/UX, HP/UX, SOLARIS, and other types of Unix systems. Supported hardware manufacturers include: IBM, DEC, Data General, Hewlett-Packard, SUN, ICL, and Sequent. Because of the large number of different systems that run UNIX or UNIX-like operating systems, the MSM system has not been tested on all possible types.

Most of these configurations support a full complement of peripheral devices including floppy disks, hard disks, streaming tape, 9-track tape, and so on. The minimum system configuration required to run MSM is a processor with 32 MB of memory; at least one disk drive; and a floppy disk, magnetic tape, or other type of removable media. Optionally, the system may include additional memory, one or more additional hard disks, and other peripheral devices.

Memory requirements

At a minimum, the MSM interpreter, database supervisor, and operating system require approximately 2 megabytes of memory. In addition, a minimum of 512K bytes of shared memory is required for terminal buffers, disk buffer pool, and other system tables required by MSM.

MSM also needs a minimum of 64K bytes per partition when the default system values are used. One partition each is required for the system, each user that logs on to the system, and each background job. The MSM system can logically support up to 8192 partitions executing concurrently on the UNIX system.

System memory requirements can change significantly when the system configuration parameters such as stap size and stack size are changed from their default values. This also is true when the default partition size is changed. Refer to “Generating the System” in this manual for additional information on these and other system configuration parameters.

Disk space requirements

The MSM system can operate with virtually any disk drive that is supported by the UNIX host operating system. To install MSM-UNIX requires a minimum of 8 MB of disk space for the Manager’s routines and globals. Additional disk space for system applications, user applications, and so on, is usually required. The exact amount of space can be configured based on the user’s actual requirements. The maximum disk space supported by a single MSM system is 512 GB.

Pre-Installation Tasks

Before you install the MSM system, perform the following preparatory tasks:

- If your distribution medium is diskettes, write-protect the original diskettes and then copy each diskette. Label the copies (referred to as the working copy) and store the original diskettes in a safe place. Use the working copy of the diskettes to install MSM.
- Back up the MSM system.

  If you are upgrading to a new version of MSM, back up the MSM database and executable files before you begin the upgrade. You can use the backup utility of your choice.
- Review the README.TXT file.

  The README.TXT file contains important information about the system that was not available when this document was printed. It includes information on new features, bug fixes, and other changes that have been incorporated since the last minor release of MSM. This documentation can be printed using the standard UNIX lp command.

Installing the MSM System

Overview

This chapter describes how to install MSM for the first time and how to update an existing MSM system to a new software release.

Distribution Package

The MSM distribution package includes all software and documentation necessary to install MSM on a UNIX-based computer. Using the supplied materials, MSM can be installed from floppy disk or magnetic tape.

MSM-UNIX Distribution Package

Item	Description
Distribution software on diskettes or tape	Contains MSM-UNIX, MSM-NET, and MSM-LAT system software in compressed format, MSM installation software, and any optional packages you purchased.
License Agreement	Contains Terms and Conditions for use of the software and the Activation Code for the license.
README.TXT	Contains late-breaking information on installation and other issues.
MSM Release Notes	Describes fixes, changes, and enhancements made to MSM since the last software release.
MSM-UNIX System Manager's Guide (this manual)	Describes features that are specific to MSM-UNIX.
MSM User's Guide	Explains how to use MSM, including logon and logoff, editing programs, peripheral devices, and so on.
MSM Language Reference Manual	Describes the M programming language as it is implemented in MSM.
MSM Utility Manual	Describes all utility programs in the MSM system.
MSM Pocket Guide	Summarizes MSM's important features.
MSM-Activate User's Guide	Explains how to use MSM-Activate.
MSM-XCALL User's Guide	Describes how to use MSM-XCALL.

Distribution Package System Files

The following table lists and describes the MSM system files that are provided with your distribution package.

Distribution Package Files

Distribution File	Description of Contents
README.TXT	A text file that describes the enhancements and fixes made to MSM-UNIX since the last printing of the documentation and the *MSM Release Notes*.
SETMODES	Invoked by the installation process to set appropriate ownership and access permissions for the distribution files.
UTILS.MGR	System-supplied utility programs that are loaded into the Manager's UCI.
GLOBALS.MSM	System-supplied global files that are loaded into the Manager's UCI.
CUAEDIT.MSM	Global files used by the CUA editor.
config.ins	A file containing system startup parameters used by the MSM installation procedure.
install	The shell script used to invoke the MSM installation process.
install.db	The database used by the MSM installation process.
license.msm	MSM licensing information.
msm/MSM	The MSM module invoked to log on to the system.
msmstart/MSMSTART	The UNIX shell script which starts MSM-UNIX.
mubij	A file that controls all input/output operations to the Before-Image Journal files.
muctrl	The MSM load module responsible for logon, logout, and system shutdown.
mudasd	The MSM database supervisor which controls all disk reads and writes to the database.
mudump	Outputs a diagnostic dump of the MSM shared memory segment.
muinit	Invoked by MSMSTART to initialize MSM-UNIX.
mujrnl	Controls all input/output operations to the After-Image Journal files.
mumsm	The MSM load module which includes the language compiler and operating system monitor.
musave	Writes modified data to disk during system shutdown.
muserver	The UNIX operating system interface module. It includes the Host File Server, tape I/O, and UNIX spooling.
mutio	Controls reads and writes to terminal type devices, including printers and modems.
sumsm	Single-user MSM-UNIX. This file is used only by the MSM installation process.

The Paper Key

When installing an MSM system, the MSM software must be activated for use through a paper key. The key consists of an activation code, the name of the supplier from whom the software was purchased, and an end-user name. The activation code is computed by an algorithm and includes the software serial number, license number, options purchased, machine type, and so on. This sequence of numbers and letters is provided on a self-adhesive label and must be entered exactly as it appears in order to enable the software.

If you move the Micronetics programs to another computer or replace the hard disk, the paper key will be needed to activate the MSM software on the new system. The paper key should be retained and is designed to be inserted into the *MSM System Manager's Guide* for safekeeping. Alternatively, the paper key can be affixed to the front or side of your computer for easy reference.

Prior to Version 4.0, the MSM system serial number and the software options purchased by the customer were stamped into the software and were contained on the system diskettes. As of Version 4.0, all optional software, such as MSM-NET and MSM-LAT, is included in every software package, but is only enabled by the activation code if the customer purchased the option.

The MSM installation process prompts you to enter information from the paper key. After you enter the information, the installation process continues. The paper key can be updated if necessary via the MSMKEY utility, or the /key startup option, which is discussed in “MSM System Operations” in this manual.

Installing MSM-UNIX

Before you begin installing MSM, carefully review the complete installation procedure.

Before you upgrade an existing MSM system to a new release, take a complete backup (MSM software modules and MSM database) of the current system. Refer to "MSM System Operation" in this manual for detailed information on performing a complete system backup.

Store the paper key in a safe location. The activation code will be required if you ever reinstall the software.

To install the MSM system, boot the computer in the normal manner following the instructions supplied by the manufacturer and then login to the root userid. If necessary, use the make directory command (mkdir) to create a sub-directory where the MSM software and database will reside. Use the change directory command (cd) to switch to the directory where MSM is to be installed.

Once you are in the proper directory, use the UNIX tar command to copy the distribution system into the directory. The format of the tar command varies depending on the hardware configuration that you are using and the version of UNIX that is installed. The following are examples of typical tar commands that could be used to load the distribution system software. The first example uses the default tape device, while the second example specifically names the tape device that is to be used.

```
tar xv
       or
tar xvf /dev/rct0
```

In the first example above, the distribution software is read from the default device that is assigned to the tar command by the Unix system. In the second example, the distribution software is read in raw mode from magnetic cartridge tape device 0. For all systems, the distribution files always are stored independently in tar format, whether the distribution is on floppy disk, cartridge tape, or 9-track magnetic tape.

When loading the MSM distribution files from floppy diskettes, you may need to load each diskette separately (issue a separate tar command for each diskette). On some systems, the tar utility allows the distribution files to span multiple diskettes. In this case, the tar utility itself prompts you to insert the second diskette at the appropriate time. On other systems, the tar command terminates without any message after the last file on the first diskette has been loaded. You must put the second diskette in the drive and repeat the same tar command used for the first diskette.

The following table lists common tar device names for many of the systems supported by MSM-UNIX. Refer to the UNIX documentation supplied with the system for additional information on the tar command and the appropriate device name for the system being used. Refer to "Tape Devices" in this manual for additional information on magnetic tape device names (cartridge and 9-track) for systems that are supported by MSM.

Tar Command Device Names

Model Name	Tar Device Name	Media Type
DG AViiON	/dev/rmt/0 (default)	Cartridge tape
HP 700 series	/dev/rmt/0m (default)	9-track tape
IBM RS/6000	/dev/rfd0	3.5-inch HD diskette
Siemens MX300/MX500	/dev/rmt/rts0 (default)	Cartridge tape
SCO UNIX	/dev/rfd0	3.5-inch HD diskette

After the distribution files have been restored, invoke the install command (enter ./install) to begin the installation procedure. You will be asked for the location of the database, its size, and the system name. After you answer the questions, verify that the information you supplied is correct. If you indicate that it is not, the installation process will be terminated and you may start the ./install again.

After the installation process begins, you will not be asked any other questions. The system automatically performs all of the necessary steps. The amount of time required to complete the installation varies, based on the hardware configuration, the size of the database being created, and the type of software that is being loaded. However, the installation process should take no more than a few minutes.

After the installation completes, use the SYSGEN utility to tailor the system to site-specific requirements. Refer to "Generating the System" in this manual for information on SYSGEN. When the completes, the system will contain the Manager's UCI (MGR), and a default programmer access code of XXX will have been assigned.

In the following instructions, the MSM-UNIX distribution tape is assumed to be loaded in the default tape drive. If this is not the case, use the f parameter on the tar command in order to specify the appropriate tape device name.

Install MSM-UNIX

1. Load the distribution tape or distribution diskette 1 in the tape drive or floppy disk drive, respectively, from which you are installing the software.
2. Log on to the root userid on your system.
3. If necessary, use the **make directory** command (mkdir) to create the subdirectory to which the MSM-UNIX executable files are to be installed.
4. Use the **change directory** command (cd) to switch to the directory in which MSM-UNIX is to be installed.
5. To copy the distribution system into the installation subdirectory, use the appropriate UNIX **tar** command for your system and tar device. For example:

   `tar xv` (to load the software from the default tape device)
6. At the # prompt, enter **./install** to begin the installation procedure.
7. Enter the activation code, supplier name, and end-user name that are provided on the paper activation key. Enter the information exactly as it appears on the paper key.
8. On the next screen, enter the location where the MSM database is to be created during the installation.
9. Enter the size (in megabytes) of the MSM database that is to be created by the installation process, for example: 10M. Use the abbreviation M for megabytes.

10. Enter the system or node name for the MSM system you are installing. Use any three uppercase letters. If you plan to use networking, the system name must be unique across the entire network.
11. Use the next screen to verify that the parameters you entered are correct.
    - To accept the parameters and install the system, enter **Y**.
    - To exit the installation process, enter **N**. To restart the installation process, enter **./install**. Enter the correct parameters using steps 7-10 above.
12. If you are installing from diskettes, the installation program prompts you to switch diskettes after the files on distribution diskette 1 are installed. Remove diskette 1 from the floppy disk drive and replace it with diskette 2 when you are prompted to do so.

    On certain systems, the tar command may finish without asking for the next diskette. If this occurs, insert the next diskette in the drive and then retype the tar command.
13. After the installation is complete, the system returns to the # prompt.

Startup of MSM-UNIX

The installation program creates a default (DFLT) configuration that enables you to start MSM-UNIX. To configure the system for your specific needs, use the SYSGEN utility. Refer to “Generating the System” in this manual for information on using SYSGEN.

Start MSM-UNIX using the default configuration

1. At the system prompt, enter **./msmstart /autoconfig**.
2. After you receive the message "MSM startup is complete," log on by entering **./msm**.
3. Enter the UCI name and the programmer access code (MGR:XXX) in uppercase characters. As a security precaution, an asterisk is displayed for each character you enter.

After you enter this information, MSM displays the job number and prompts you with the > symbol. You now are logged on to MSM in programmer mode.

Updating MSM to a New Release

From time to time, you will receive new releases of the MSM system. New releases generally are supplied whenever enhancements, new utilities, or corrections for errors are made to the system. This typically occurs one or two times a year.

A new paper key is not needed to activate a new maintenance release (for example: Version 4.01, 4.02). However, a new key will be supplied to the customer when a new MSM version is released (Version 4.0, 5.1, and so on).

The procedure for updating to a new release of MSM is virtually identical to the installation process described in the previous sections. To perform an update, use the following steps.

Perform a system update

1. Take a complete backup of the MSM database and distribution files.
2. Use the UNIX tar command to load the new MSM distribution system into the directory where you originally restored them.
3. To invoke the installation process, type **./install**. The installation process them prompts you to respond to the same questions described in the installation sections above.

The only difference is that when you specify the UNIX path name, the installation process detects that the M database already exists. Specify the same directory name that you supplied when you performed the initial install that created the M database.

Note When a new version of MSM includes major interpreter changes, you must run the %RELOAD utility after completing the update. Failure to do so will result in <PLDER> errors when routines are accessed. Refer to the *MSM Utility Manual* for additional information on the %RELOAD utility.

When you update a system, information in the existing database is retained. Only the new utilities are installed. After you answer the installation questions, verify that the information you supplied is correct. If you indicate that it is not, the update process is terminated.

The CONFIG.msm Parameter File

During installation, MSM creates a CONFIG.msm file in the directory you specified for the database location. This file contains information about your MSM system and is used during startup to determine how MSM is to be set up.

The first line of the CONFIG.msm file contains the fully-qualified file name for the first volume of the first Volume Group that is to be mounted. This file is sometimes referred to as the *boot* or *startup database*. The boot database created by the MSM installation procedure is called database.msm.

If you move the boot database to a different location, you must edit the CONFIG.msm file and update the path name to the database.

The second line of the CONFIG.msm file contains a series of parameters used by MSM during the startup process. These parameters are maintained by the SYSGEN utility and should not be modified manually.

If the startup fails and the error message “shmget”, “shmat”, “semget”, or “no space left on device” occurs in the msmlog file, then use the –b or –t startup options to override the CONFIG.msm settings. Refer to “MSM System Operation” in this manual for additional information on startup options.

At system startup, MSM attempts to access the CONFIG.msm file in the directory from which the msmstart command was invoked. If it does not find it there, then MSM searches the /etc directory. If the file is not found in either location, an error message is issued and the startup is terminated.

Generating the System

This chapter describes the process for generating the MSM system and tailoring it to the specific hardware configuration and operating environment where it will be installed.

Generating the MSM System

Use the system generation process (SYSGEN) to define one or more hardware and software configurations to be used by MSM-UNIX. A *configuration* is a collection of attributes (for example: device types, terminal port definitions, tied terminal indexes, number of partitions, and so on) that describe the operating environment for MSM. Multiple configurations are permitted, in which each definition has a unique name.

The basic elements of a configuration include definitions for tied terminals, terminal ports, basic system parameters, and so on. The result of the SYSGEN process is a configuration definition which is stored in the SYS global of the Manager's UCI. The system startup routine (STU) uses the configuration descriptions to initialize the system and configure it with the defined parameters. Refer to "MSM System Operation" in this manual for information on how the SYSGEN configuration is used during system startup and how to override individual parameters on the startup command.

Before you begin the SYSGEN process for a new system, define a list of the UCIs to be created. You also should define disk information associated with the UCIs (for example: location for routine and global directory, expansion area for pointer blocks, expansion area for data blocks, and so on).

Define hardware-specific information for the configuration, including tape device names, network adapter device names, and serial port device names which are to be defined in MSM. System parameters for options such as backspace character, line delete character, and number of partitions must be determined.

The SYSGEN Process

The SYSGEN is performed as an interactive dialogue with the system. The system prompts you when information is required.

At any prompt, enter **?** to receive help information or **^L** to obtain a list of the current values that are defined for the prompt. If you enter a null response (a carriage return), the original value is left unchanged. You can use null responses and ^L to display but not change items in a configuration. To return to the previous question at any time during the SYSGEN, enter ^ (circumflex) at any prompt.

In the SYSGEN process, the order in which the configuration is specified is important because of inter-relationships between configuration elements. The following configuration elements are related:

- UCI Definitions
- Tied Terminal Table Definitions
- Port (Terminal) Definitions

For ports (terminals), the port definition includes the index into the Tied Terminal table for the routine to be executed at logon. Similarly, the tied terminal definition includes the name of the UCI where the routine that is to be executed at logon is stored. The port definition requires the tied terminal index to be previously defined, and the tied terminal index requires the UCI index to be previously defined.

Use the following steps to define a system configuration that uses tied terminals:

1. Create the volume group on which the UCI is to reside.
2. Create the new UCI.
3. Define the tied terminal entries.
4. Define the ports.

If the UCI to be created will reside on a volume group other than the volume group that is used to boot the system, you must create and mount the volume group before you can create the UCI.

Invoking the SYSGEN Utility

To perform the SYSGEN process, you must be logged on to the Manager's UCI in programmer mode. When a SYSGEN is performed in a multi-user environment, other users may be logged on to the system. Because most SYSGEN parameters are acted upon by the STU (system startup) routine, you must shut down MSM (using SSD) and then restart it before most of the new values will take effect. Some new values (for example: new UCIs) are available without restarting MSM.

The following example demonstrates how to invoke the SYSGEN utility and also illustrates the SYSGEN high-level menu.

```
>D ^SYSGEN

Select SYSGEN Option:

    1 - Display Configuration Parameters
    2 - Create New Configuration
    3 - Edit Configuration Parameters
    4 - Edit Configuration Name/Comment
    5 - Delete Configuration
    6 - Set Default Startup Configuration
    7 - UCI Management
    9 - System Configuration Parameters
   10 - Database Definition
   12 - Device Translation Tables
   13 - Mnemonic Namespaces
   14 - Journaling Management

Select Option: ?

    Select option by specifying the option number or supplying enough
    characters to identify the option,
    To get help information specific to an option, enter '?' followed
    by the option number,
    Enter <RETURN> or '^' to exit with no action,
    Enter '^L' to get a list of options.

Select Option: RETURN

>
```

When the SYSGEN process begins, the system prompts you to enter all necessary information. The following sections describe the SYSGEN menu options and provide sample terminal sessions.

Display Configuration Parameters

Use this option to display all the specified parameters for a selected configuration.

Enter the name of the configuration to be displayed. The system prompts you to enter an output device. The default device is the current terminal device

```
Select Option: 1 - Display Configuration Parameters

Select Configuration <DFLT>: RETURN

Enter Output Device <1>: RETURN
------------------------------------------------------------------
Backspace, Line Delete Character
Current Backspace Character: Ctrl-H
Current Line Delete Character: Ctrl-X
------------------------------------------------------------------
Maximum Partitions
Maximum Partition Limit: 8
------------------------------------------------------------------

     .
     .
     .

     Additional Configuration Parameters

     .
     .
     .

------------------------------------------------------------------

VIEW Command Restriction:

VIEW Command Is NOT Restricted

------------------------------------------------------------------
LOCK Table Size
Maximum LOCK Table Size: 2K
------------------------------------------------------------------


Select Option:
```

Create New Configuration

Use this option to create a new system configuration and initialize it with a set of default values.

Enter the name of the configuration to be created, a descriptive comment to be stored with the configuration name, and the name of the configuration from which the default values are to be copied.

After you create a new configuration, use the Edit Configuration Parameters option on the SYSGEN main menu to update the values. If you enter a null response to the "Enter Name of New Configuration" prompt, the system returns to the main option menu.

```
Select Option: 2 - Create New Configuration

Enter name of new configuration: ?

    Select a configuration to be edited (or created) by specifying its
    full name.  A configuration name may be any combination of letters
    and/or numbers with a maximum length of 14 characters.
    Enter '^L' to list the names of current configurations.

Enter name of new configuration: TEST

Enter comment: Test Configuration

Enter name of configuration to copy from: ?

    If you want the new configuration to be a duplicate of an existing
    configuration, enter the name of the configuration. All entries
    will be copied to the new configuration. If you do not enter a name
    here, the new configuration will contain only a minimum set of
    default values.   Enter '^L' to list the names of existing
    configurations.
    Enter name of configuration to copy from: ^L

Current Configuration: MDC

Available Configurations:
  Id: DFLT.............02/14/98: DEFAULT CONFIGURATION
  Id: MDC..............02/23/98: Production System

Enter name of configuration to copy from: DFLT

The new configuration is defined as follows:

Backspace, Line Delete Character

Current Backspace Character: Ctrl-H

Current Line Delete Character: Ctrl-X
-----------------------------------------------------------------

     .
     .
     .

     Additional Configuration Parameters

Select Option:
```

Edit Configuration Parameters

The system manager does not need to do a complete SYSGEN to change individual system parameters. The Edit Configuration Parameters option allows individual parameters for an existing configuration to be displayed, modified, or deleted; also, new entries can be added to the configuration.

The basic system parameters are divided into logical groupings. Each group appears as an entry on the Edit Configuration Parameters menu. Some groups contain sub-menus that allow multiple functions (for example: edit, list, delete) to be performed. Following is a list of these groupings.

Full SYSGEN

Backspace and Line Delete Character *

Automounts and Autostarts *

Maximum Partitions *

Programmer Access Code

MSM Disk Usage *

VIEW Command Restriction

Tied Terminal Table

Port (Terminal) Definition *

External Call Configuration *

Default Partition Size

Translation/Replication Table

DDP and LAT Configuration *

Tape Device Definition *

Global Defaults *

LOCK Table Size *

Display Configuration Parameters

Mode Flags *

When any parameter marked with an asterisk (*) in the list above is modified, the change does not take effect until the STU routine is run. Shut down and then restart the system after changing these parameters. Refer to “MSM System Operation” in this manual for additional information on system startup and shutdown.

When the Edit Configuration Parameters option is selected, a sub-menu is displayed. The sub-menu provides access to each logical group of system parameters associated with a configuration. The following sections provide an example of invoking this menu and a detailed description of each parameter. When you select this option, MSM prompts you for the name of the configuration to edit. If no configuration name is supplied, the system returns to the main menu.

```
Select Option: 3 - Edit Configuration Parameters

Select Configuration <DFLT>: ?

    Select a configuration to be edited/created by specifying its full
    name. A configuration name may be a combination of letters and/or
    numbers with a max length of 14 characters.
    Enter '^L' to list the names of current configurations.

Select Configuration <DFLT> ^L

Available Configurations:

  Id: DFLT.............01/04/98: DEFAULT CONFIGURATION

Select Configuration <DFLT>: DFLT

Select SYSGEN Option:

       1 - SYSGEN (step through full SYSGEN)
       2 - Backspace, Line Delete Character
       3 - Automounts and Autostarts
       4 - Maximum Partitions
       5 - Security
       6 – Tuning Parameters
       8 - Tied Terminal Table
       9 - Port (Terminal) Definition
      11 - External Calls Configuration
      12 - Default Partition Size
      13 - Translation/Replication Table Maintenance
      15 - DDP and LAT Configuration
      16 - Tape Device Definition
      17 - Global Defaults
      18 - LOCK Table Size
      19 - Display Configuration Parameters
      20 - Mode Flags

    Select Option:
```

SYSGEN (Step Through Full SYSGEN)

Use this option to perform a complete SYSGEN. Following is an example of the text that will be displayed.

```
Select Option: 1 - SYSGEN (step through full SYSGEN)

Would you like an explanation of the SYSGEN <N>: Y

    The SYSGEN process allows you to define specific configuration to
    be used by MSM.  The configuration includes definitions for
    character delete (backspace) and line delete, and ports. You will
    be prompted for each definition.
    Respond with a '?' to any of the prompts for help information.
    A response of '^L' displays current values when multiple
    definitions are possible.

Note: A <RETURN> response will exit the current question with no other
action being performed.  Thus you may use the <RETURN>and '^L' to
display the current configuration without redefining it. The <RETURN>
is also used to back up to the previous prompt whenever you are in the
middle of redefining a specific element of the configuration.
The order of specifying a configuration is important due to the inter-
relationships between elements of a configuration.  The following are
the inter-related elements of a configuration:

  - Port definitions
  - UCI definitions
  - Tied Terminal Table
```

```
   The definition of a port includes the specification of a routine to
   be invoked as part of the logon (via Tied Terminal Table).  The
   Tied Terminal definition in turn specifies the UCI name for logging
   on.  The Port definition requires that the corresponding Tied
   Terminal Index be already defined.  The Tied Terminal definition
   requires that the UCI be defined to the system. Thus, the order of
   configuring the system must be:

 1 - Define the volume group on which the UCI is to reside
 2 - Create the UCI (if needed)
 3 - Define the Tied Terminal Table entries
 4 - Define the ports
```

If the UCI(s) to be created will reside on a volume which was not defined when the system was booted, perform the following steps:

```
 1 - Create the database volume
 2 - Shut down the system (DO ^SSD)
 3 - Reboot
 4 - Repeat the SYSGEN process (DO ^SYSGEN)
```

At this point, the system begins to step through the entire SYSGEN process. Options will be presented in the order shown in the sample text above.

Backspace and Line Delete Characters

The backspace character indicates to MSM-UNIX that the previously entered character is to be discarded (thrown away and not passed on to the program issuing the READ operation). When entered in response to a READ command, MSM erases the character from the terminal if it is a CRT. If it is not a CRT, then MSM will echo a slash (/) character.

Similar to the backspace character, the line delete character indicates to MSM-UNIX that all previously entered characters are to be discarded (all characters entered in response to the READ operation are thrown away). When entered in response to a READ command, MSM erases the characters that have been entered so far and waits for the user to re-enter the entire line. The line delete function is inhibited during single character reads (READ *X).

MSM allows the ASCII backspace ($CHAR(8)), the ASCII delete ($CHAR(127)), or both to be the system backspace character. MSM handles the line delete character in much the same way as the backspace character. Valid characters are CTRL/X, CTRL/U, or both.

In a mixed-terminal environment, you need not be concerned with which character is generated by a given terminal. In response to the backspace character prompt and the line delete prompt, enter the actual character that is to represent the function, or enter the letter B to indicate that both characters are to be recognized.

Whenever the defined backspace character is the DEL character, the system treats the true ASCII backspace character as an ordinary character. In this case, even though the cursor is physically moved back one space, the previous character is retained. This feature is useful for over-striking.

The backspace and line delete characters are treated as normal characters during single character READ (READ *X) operations. This means that the system takes no special action when either the backspace or line delete character is entered. On single character READs, the system returns the actual ASCII value of the character that is entered.

```
Select Option: 2 - Backspace, Line Delete Character

Current Backspace Character(s): Ctrl-H

New Backspace Character: ?

    Enter the actual character you want to be recognized as the
    'backspace' character, either backspace (Ctrl-H) or DEL (rubout) or
    'B' for both.

Current Backspace Character(s): Ctrl-H

New Backspace Character: B   Ctrl-H, DEL

Current Line Delete Character: Ctrl-X

New Line Delete Character: ?

    Enter the actual character you want to be recognized as the 'line
    delete' character, either Ctrl-U or Ctrl-X or 'B' for both.

Current Line Delete Character: Ctrl-X

New Line Delete Character: B   Ctrl-U, Ctrl-X

Select Option:
```

Automounts and Autostarts

Use this option to specify which volume groups are to be automatically mounted by MSM and whether spooling, journaling, and other services are to be automatically started.

Invoking the Automounts and Autostarts option

When you select this option, the system displays the following sub-menu. The following sections include sample terminal sessions for each menu option.

```
Select Option: 3 - Automounts And Autostarts

Select Autostart Options

     1 - Volume Group Automount
     2 - Spooling Autostart
     3 - Journal Autostart
     4 - Cross-System Journal Autostart
     5 - Automatic Partition Startup

Select Option:
```

Volume Group Automount

This option maintains a table of volume groups to be automatically mounted during system startup. The table consists of a set of indexes, each of which describes the file name of the first volume of the volume group to be mounted. For Remote Volume Groups, the system and volume group names are entered in place of the file name. At startup, the STU routine reads the table and mounts each specified volume group.

Although indexes can be from 1 to 31 and can be added in any order, volume groups are mounted in strict numerical order beginning with 1. Assume that indexes 1 through 4 are added to the table, and subsequently, indexes 1 and 3 are deleted. At system startup, the volume group associated with index 2 is mounted as volume group 1, and the volume group associated with index 4 is mounted as volume group 2.

```
Select Option: 1 - Volume Group Automount

Enter Index Number: ?

    Enter an index number in the automount table (1 to 31) which will
    define a volume group to be mounted during startup.
    These volume groups will be mounted in the order in which they
    appear in this table.
    Enter ^L to list the current automount table.

Enter Index Number: ^L

Index  Type    Access  LAT  Host File Name or Remote Id
-----  ------  ------  ---  --------------------------
  1    Local     -      -   /msmjrnl/journal.msm
  2    Local     -      -   /usr/msm/dmstest.msm

Enter Index Number: 3

Host File Name: /disk2/mumps/account.db

Volume Group Type <Local>: ?

    Enter 'LOCAL' if the volume group is local to this system.
    Enter 'REMOTE' if the volume group is mounted on a remote system.
    Enter '^' to return to the previous question.

Volume Group Type <Local>: RETURN

Enter Index Number: 4

Volume Group Type <Local>: REMOTE

System,Volume Group: ?

    Enter the remote system name, and the volume group name,
    separated by a comma. All names must be three uppercase letters.
    Enter '^' to return to the previous question.
    Enter '^L' to list the current automount entries.

System,Volume Group: FSA,FDA

Override default access <N>: ?

    Enter NO to use the default access taken from the volume group
    characteristics.
    Enter YES to specify the access (i.e., READ/WRITE access, and LAT
    availability) requested for this mount.

Override default access <N>: RETURN

Enter Index Number: RETURN

Select Option:
```

Spooling Autostart

Use this option to specify whether or not the Spooling function is to be automatically invoked when the MSM system is started. The following terminal session illustrates this function.

```
Select Option: 2 - Spooling Autostart

Auto Enable Spooling <N> ?

    Enter Y or N to autostart spooling. The spool area must be defined
    for this option to work.

Auto Enable Spooling <N> Y

Enter a spool device: ?

    Select a device to be used as the virtual spool device. Any output
    to this device will not print immediately, but will create a spool
    file to be printed in turn by the despooler. If you do not enter a
    number, device 2 (actual  spool device) is the only device which
    can be used to create spool files.
    Enter '^' to return to the previous question.
    Enter '^Q' to exit this utility.

Enter a spool device: 3

Do you want to start the despooler <YES> ?

    Enter 'YES' to start the despooler
    Enter '^' for previous question, '^Q' to quit.

Do you want to start the despooler <YES> YES

Enter the device to despool to <3> ?

    Enter the number of the device to despool to.
    Enter '^' to return to the previous question.
    Enter '^Q' to exit this utility.

Enter the device to despool to <3> RETURN

Select Option:
```

Journal Autostart

Use this option to specify whether or not the Journaling function is to be automatically invoked when MSM is started. The following terminal session illustrates the Journaling Autostart function.

```
Select Option: 3 - Journal Autostart

Autostart Journaling <N>: ?

    Enter Y or N to Autostart Journaling. You must have available
    journal spaces before Journaling can be automatically started.

Autostart Journaling <N>: Y
```

Cross-System Journal Autostart

Use this option to specify whether or not the Cross-System Journaling feature of MSM is to be automatically started when the system is started. The following terminal session illustrates this function.

```
Select Option: 4 - Cross-System Journal Autostart

Autostart Cross-System Journaling <N>: ?

    Enter YES to automatically start Cross-System Journaling during
    system startup.

Autostart Cross-System Journaling <N>: Y
```

Automatic Partition Startup

Partitions may be automatically activated as part of the system startup process. The entries in the partition startup table are used as operands of the JOB command. Each entry consists of a numeric index (which need not be sequential), an entry reference (a routine name or a line reference with a routine name), a UCI name where the routine is to be started, and as an option, the partition size required for the job. The partitions are started in the order of the numeric indexes.

```
Select Option: 5 - Automatic Partition Startup

Current Automatic Startup List:

          Index    UCI    Psize   Entry reference
          -----  -------  ------  ---------------
            1      MGR      10    ^MONITOR

Select index: ?

    Specify the index number of an entry to be added or changed.
    Partitions are started in numeric order according to the index.
    To delete an entry, specify a '-' in front of the number.
    To list existing entries, enter '^L'.

Select index: 2

Enter UCI name: ?

    Specify the UCI name under which the partition will execute. A UCI
    name consists of three uppercase letters. If the UCI is on a
    secondary volume group, then enter UCN,VGN, where UCN is the UCI
    name and VGN is the volume group name.
    Enter '^L' for a list of UCIs on all mounted volume groups.

Enter UCI name: MGR

Enter Entry Reference: ?

    Specify an entry reference of where execution should begin.
    Valid syntax for an entry reference is:

         ^routine name
         +offset^routine name
         label^routine name
         label+offset^routine name

Enter Entry Reference: ^%ZTM
```

```
Enter partition size <SYSTEM>: ?

    Specify the partition size for the job, or 'SYSTEM' if the job
    should use the system default partition size.

    The maximum value for a partition size is 256.

Enter partition size <SYSTEM>: RETURN

Select index:
```

DDP autostart

Use this option to specify whether or not the Distributed Data Processing (DDP) feature of MSM is to be automatically started when the system is started.

```
Select Option: 6 - DDP Autostart

Autostart DDP <NO>: ?

    Enter YES to start up DDP (DDPSRV) during system startup.
    Enter NO to require manual startup via the ^DDP utility.

Autostart DDP <NO>: Y

Select Autostart Options:
```

Workstation server autostart

Use this option to specify whether or not the MSM-Workstation server utility is to be automatically started when the system is started.

```
Select Option: 7 - Workstation Server Autostart

Autostart Workstation Server <NO>: ?

    Enter YES to start the Workstation server utility (MSERVER) during
    system startup.
    Enter NO to require manual startup via the following command:
    JOB START^MSERVER

Autostart Workstation Server <NO>: Y

Select Autostart Options:
```

LAT autostart

Use this option to specify whether or not MSM-LAT is to be automatically started when the system is started.

```
Select Option: 8 - LAT Autostart

Autostart LAT <NO>: ?

    Enter YES to start up LAT (DDPADV) during system startup.
    Enter NO to require manual startup via the ^LAT utility.

Autostart LAT <NO>: Y

Select Autostart Options:
```

Telnet service autostart

Use this option to specify whether or not the MSM Telnet service is to be automatically started when the system is started.

```
Select Option: 9 - Telnet Service Autostart

Autostart Telnet Service <NO>: ?

    Enter YES to start the MSM telnet service during system startup.
    Enter NO to require manual startup via DO ENABLE^TELNET.

Autostart Telnet Service <NO>: Y

Select Autostart Options:
```

MSM-Activate service autostart

Use this option to specify whether or not the MSM-Activate service is to be automatically started when the system is started.

```
Select Option: 10 – MSM-Activate Service Autostart

Autostart MSM-Activate Service <NO>: ?

    Enter YES to start the MSM-Activate service during system startup.
    Enter NO to require manual startup via DO ^APIMGR.

Autostart MSM-Activate Service <NO>: Y

Select Autostart Options:
```

PDQweb service autostart

Use this option to specify whether or not the MSM-PDQweb service is to be automatically started when the system is started.

```
Select Option: 11 - PDQweb Service Autostart

Autostart PDQweb Service <NO>: ?

    Enter YES to start the MSM-PDQweb service during system startup.
    Enter NO to require manual startup via DO START^MSERVER("%MSMWEB").

Autostart PDQweb Service <NO>: Y
```

Server management autostart

Use this option to specify whether or not Server Management is to be automatically started when the system is started.

```
Select Option: 12 –Server Management Autostart

Autostart Server Management <NO>: ?

    Enter YES to automatically start Server Management during system
    startup.

Autostart Server Management <NO>: Y
```

Maximum Partitions

Use the Maximum Partitions option to specify the number of concurrent tasks and the maximum number of users that can be active on the system. The number you can specify depends on the limits of your MSM license. Once the maximum number of active partitions is reached, no additional users can log on and no additional background jobs can be started. If the maximum number of users has been reached but the maximum number of partitions has not been reached, then additional background jobs can be started.

When MSM is first installed, this option is set to eight partitions and eight users. Before increasing these values, verify that the UNIX semaphore configuration is appropriate for the number of users and ports. Refer to "Configuring UNIX" in this manual for information on semaphore configuration.

In theory, the number of partitions that can be active at one time also is limited by the amount of system memory available to MSM. If there is insufficient memory to create a new partition, new users are prevented from logging on to the system. Any attempt to create a background job will hang until memory becomes available. Because UNIX is a virtual-memory operating system, it pages memory to disk rather than reporting an out-of-memory condition. Overall system performance could be drastically affected if the system is memory-constrained and UNIX begins to page memory to disk.

```
Select Option: 4 - Maximum Partitions

Maximum number of partitions <8>: ?

    Enter the maximum number of concurrently active partitions allowed
    for this configuration.
    The value cannot exceed the limits of the license.

Maximum number of partition <8>: 48

Maximum number of users <8>: ?

    Enter any value up to the limit of the license.  MSM will limit the
    maximum number of concurrent users to the specified value.

    One can set the value lower than the limit of the license for the
    purpose of decreasing the number of network connections used on any
    server that this system connects to.

Maximum number of users <8>: 32

Select Option:
```

Security

Use this option to manage security features that restrict user access to specific system functions. When you select this option, the system displays a sub-menu of options.

Use Option 1, Programmer Access Code (PAC), to control access to programmer mode in MSM. When a PAC is specified, users must enter the correct PAC before they can log on to the system in programmer mode. If the correct PAC is not entered, access to the system is denied.

To specify a PAC, enter the actual value of the code at the Programmer Access Code prompt. The code may be from one to 12 characters in length and may include any valid ASCII character except the reserved control characters, the colon (:) character, and the comma (,) character. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for a list of the reserved control characters. During system installation, a default code of XXX (three uppercase Xs) is created.

Use Option 2, VIEW Command Restriction, to restrict use of the VIEW command to a program executing in the Manager's UCI, a percent routine, or a routine that is called from a percent routine. Any other use of the VIEW command causes a <PROT> error. Although % routines are stored in the Manager's UCI, they can be invoked from any UCI. Because the VIEW command is potentially dangerous if it is used incorrectly or if users are not well-versed in the internal structure of MSM control or disk blocks, use of this option may be restricted to the system manager.

Use Option 3, Run Mode Security, to designate which routines may be accessed in run mode, directly from the UCI login prompt. When this option is disabled, any routine can be accessed in run mode. Optionally, you may assign a password that must be entered in order to invoke a specific routine in run mode.

Invoking the Security Option

When the Security option is selected, the system displays the following sub-menu. The terminal sessions in the following sections demonstrate use of each menu option.

```
Select Option: 5 - Security

Available Functions:

   1 - Programmer Access Code
   2 - VIEW Command Restriction
   3 - Run Mode Security
```

Programmer Access Code (PAC),

Use this option to restrict user access to programmer mode.

```
Select Option: 1 - Programmer Access Code

Current Programmer Access Code: XXX

New Programmer Access Code: ?

    Enter the actual Program Access Code (1-12 characters) you wish MSM
    to recognize for logging on to UCIs.

Current Programmer Access Code: XXX

New Programmer Access Code: VGH

Select Option:
```

VIEW command restriction

Use this option to limit use of the VIEW command to the Manager's UCI and % routines.

```
Select Option: 2 - VIEW Command Restriction

Restrict use of VIEW command to Manager's UCI? <NO>: ?

    Enter YES to restrict use of the VIEW command to the Manager's UCI
    and '%' routines only. If this option is turned on, any VIEW
    command which is encountered outside of the Manager's UCI which is
    not a part of a '%' routine will generate a <PROT> error.
    Enter NO to allow the VIEW command from any UCI.
    Enter '^' or '^Q' to exit this option."

Restrict use of VIEW command to Manager's UCI? <NO>: RETURN

Select Option:
```

Run mode security

Use this option to restrict user access to routines in run mode.

```
Select Option: 3 - Run Mode Security

Enable Run Mode Security <NO>: ?

    If Run Mode Security is enabled, then only those routines defined
    through this option can be accessed in run mode, directly from the
    UCI login prompt. If Run Mode Security is disabled, any routine can
    be accessed in run mode.
    Enter ^ to exit this option.

Enable Run Mode Security <NO>: YES

Select UCI: ?

    Enter the UCI,VGN in which the routine to be enabled resides.
    The volume group name (VGN) is optional, and defaults to the boot
    volume group.
    To enable a routine for ALL UCIs, enter an asterisk (*).
    Enter '^' to return to the previous prompt.

Select UCI: MGR

Enter routine name: ?

    Enter the name of a routine residing in the UCI that you have
    selected to which access is to be allowed in run mode (invoked
    directly from the login UCI prompt). Optionally, these routines can
    also be password-protected.

    Enter ^L to list the routines which are currently enabled.
    Enter '-routine' to disable a routine which was previously enabled.
    Enter ^ to return to the previous prompt.

Enter routine name: ^L

Enter output device <0>: RETURN

    UCI,VGN    Routine    Password
    -------    -------    --------
    *          %GL
    MGR        SYSGEN     genit
    PRD,SYA    ZU

Enter routine name: %RR

Enter password: ?

    If a password is entered, then that password must be entered when
    invoking the routine in run mode.
    The format of the run mode login at the UCI prompt is as follows:
    UCI{,VGN}:routine{:password}
    The volume group (VGN) and the password are optional.

    If no password is entered, none will be required.
    To delete a password previously specified, enter a minus sign (-).
    Enter ^ to return to the previous prompt.

Enter password: rst
Routine ^%RR in UCI MGR with password rst added to table.

Enter routine name: ^L

Enter output device <0>: RETURN

    UCI,VGN    Routine    Password
    -------    -------    --------
    *          %GL
    MGR        %RR        rst
    MGR        SYSGEN     genit
    PRD,SYA    ZU

Enter routine name:
```

Tuning Parameters

MSM supports system parameters that can be modified to affect aspects of system performance. Although the default values of these parameters are appropriate for most systems, you may modify one or more parameters to achieve optimum performance.

The following sample terminal session includes the online help for each parameter, which should be sufficient to determine the potential effects of modifying the value. Do not change these parameters unless you identify and thoroughly understand a specific performance problem.

```
Select Option: 6 - Tuning Parameters

Disk I/O Rate:           150*
Max Modified Buffers:    256
Commands per Slice:      5000*
WaitQ2 Threshold:        50000*
WaitQ3 Threshold:        500000*

Values shown with an asterisk are system defaults

Disk I/O rate <150*>: ?

Enter the disk I/O rate for this system.

    The disk I/O rate is the maximum sustainable rate (in 1-K
    blocks/sec) at which physical disk I/O can be performed by this
    computer. If more than one disk drive is present, then the rate
    should be an approximate average of all disk subsystems.

    Setting the I/O rate too low will reduce the rate at which modified
    buffers are written to the disk. Setting the rate significantly
    higher than the disk subsystem can actually perform will tend to
    elevate the performance of jobs which are primarily updating the
    database, at the expense of interactive jobs (i.e., users).

    Changes to this value are not necessary for most systems, and
    should be made with great care, since improper values could degrade
    performance.

    Enter - to reset the value to the default for this system.
    Enter ^ to return to the previous prompt.

Disk I/O rate <150*>: 200

Maximum number of modified buffers <256>: ?

    Enter the maximum number of modified buffers to be allowed by the
    system.

    When the number of modified buffers reaches this level, any job
    which needs to modify a buffer must first wait while one modified
    buffer is written.  In this way, the performance of disk-intensive
    jobs is limited so that sufficient unmodified cache exists to
    support the needs of interactive jobs (i.e., users).

    Changes to this value are not necessary for most systems, and
    should be made with great care, since improper values could degrade
    performance.

    Enter - to reset the value to the default for this system.
    Enter ^ to return to the previous prompt.

Maximum number of modified buffers <256*>: RETURN
```

```
Commands per slice <5000*>: ?

    Enter the number of commands per command slice.

    This is the maximum number of commands that a single job will be
    allowed before it must wait for other jobs to get their turn.

    Decreasing this value tends to improve response time, but can
    result in a "jerky" appearance when displaying data on the screen.
    Increasing this value tends to improve overall throughput due to a
    decrease in job dispatch frequency. However, certain users might
    perceive a decrease in response time at specific points in the
    application.

    Changes to this value are not necessary for most systems, and
    should be made with great care, since improper values could degrade
    performance.

    Enter - to reset the value to the default for this system.
    Enter ^ to return to the previous prompt.

Commands per slice <5000*>: RETURN

WaitQ2 threshold <50000*>: ?

    Enter command threshold for placing a job in WaitQ2.

    When a job has used up its allotted command slice, it is placed in
    one of three primary wait queues, depending on how interactive it
    is.

    Interactivity is determined by the frequency of terminal reads.
    Immediately after a terminal read, jobs are assigned to reside in
    the highest priority queue, WaitQ1.  When the number of commands
    since the last terminal read exceeds the WaitQ2 threshold, but does
    not exceed the WaitQ3 threshold, then the job is placed in WaitQ2
    at the end of its command slice.

    When a job is in WaitQ2, it will not execute as long as at least
    one job resides in WaitQ1.

    Changes to this value are not necessary for most systems, and
    should be made with great care, since improper values could degrade
    performance.

    Enter - to reset the value to the default for this system.
    Enter ^ to return to the previous prompt.

WaitQ2 threshold <50000*>:

WaitQ3 threshold <500000*>: ?

    Enter command threshold for placing a job in WaitQ3.

    When a job has used up its allotted command slice, it is placed in
    one of three primary wait queues, depending on how interactive it
    is.

    Interactivity is determined by the frequency of terminal reads.
    Immediately after a terminal read, jobs are assigned to reside in
    the highest priority queue, WaitQ1.  When the number of commands
    since the last terminal read exceeds the WaitQ3 threshold, then the
    job is placed in WaitQ3 at the end of its command slice.

    When a job is in WaitQ3, it will not execute as long as at least
    one job resides in either WaitQ1 or WaitQ2.

    Changes to this value are not necessary for most systems, and
    should be made with great care, since improper values could degrade
    performance.

    Enter - to reset the value to the default for this system.
    Enter ^ to return to the previous prompt.
```

Tied Terminal Table

In MSM, the Tied Terminal table is used to link a specific terminal device to an application routine. Once a terminal has been tied to a particular application routine, it can only access that routine or routines called by the original routine. The terminal cannot be logged on to the system in programmer mode. If programming errors or interrupts occur that are not trapped by the application, the terminal is automatically logged off the system.

Before a terminal can be tied to an application routine, an entry must be created in the Tied Terminal table. Each entry in the Tied Terminal table is numbered (it has an arbitrary index value associated with it) and contains the name of the routine, the UCI where it is stored, and the partition size required by the routine. When a user attempts to log on to MSM, the system determines whether the terminal is tied to an application.

If the terminal is tied to a specific routine (application), then that routine is invoked. If the terminal is not tied, then the standard logon routine (%LOGON in the Manager's UCI) is invoked.

```
Select Option: 8 - Tied Terminal Table

Current Tied Terminal Table List:

Index  Routine Reference  UCI,VGP  Partition Size  Ports Tied To Routine
-----  -----------------  -------  --------------  ---------------------
  1    Menu^PATIENT       PRD      48              5, 7, 11, 15

Enter Tied Terminal Index: ?

    The Tied Terminal Table is composed of entries, each of which
    represents a routine to be invoked as part of the logon process.

    Select an index to this table:

      To create an entry - Specify a non-existing index (use '^L' for
                           list of presently defined entries).
      To edit an entry   - Specify the index of the desired entry.
      To delete an entry - Prefix the index of the entry to be deleted
                           with a minus ('-'); e.g.: -5  will delete
                           entry number 5.

    The ordering (value of the index) is NOT important. To 'tie' a
    specific port to a routine, create an entry in this Tied Terminal
    Table, and specify its index when defining the port.

    Routine '%LOGON' is supplied in the distribution system as the
    default logon routine.  Unless a terminal is 'tied' to a particular
    routine via the Tied Terminal Table, routine '%LOGON' will be
    invoked. This routine will leave the terminal in Programmer Mode if
    the Programmer Access Code (PAC) is supplied.

Enter Tied Terminal Index: 2

Enter Routine Reference: ?

    Enter the name of a routine to be automatically invoked when a
    login is initiated from a port with a tied terminal index of this
    entry and the terminal is not presently logged on.

    A routine name, or a full entry reference (LABEL^RTN)
    must be entered.

Enter Routine Reference: ^FINANCIAL RETURN
```

```
Select UCI: ?

    Enter the UCI and volume group in which the selected routine is to
    be invoked. If the volume group is omitted, volume group 0 will be
    used. The UCI and volume group name must be separated by a comma.

    UCI and volume group names must be 3 uppercase letters.

Select UCI: PRD RETURN

Enter Partition Size for ^FINANCIAL <48>: ?

    Enter a positive integer corresponding to the partition size in 1-K
    blocks of the partition in which this routine will run.

   Partition sizes may range from 12K to 256K.

Enter Partition Size for ^FINANCIAL <48>: RETURN

Do you wish to tie any ports to this entry <N>: ?

    You may link this entry to an existing port definition, or the
    linkage may be done when the port definition is created/edited.

Do you wish to tie any ports to this entry <N>: Y

Enter Port Number, or L (Dynamic LAT ports), or D (other Dynamic
Ports): ?

    Enter the port number which will be tied to this entry.
    Enter "L" to tie all dynamic LAT ports to this entry.
    Enter "-L" to untie dynamic LAT ports from this entry.
    Enter "D" to tie all dynamic ports to this entry.
    Enter "-D" to untie dynamic ports to this entry.

    Dynamic ports are those which are not defined in SYSGEN, but are
    nevertheless allowed to login via LAT, TCP/IP, or serial
    connection, by virtue of an available Device Descriptor Block
    (DDB). Dynamic non-LAT DDBs are created whenever the maximum number
    of devices exceeds the number of non-LAT ports defined in SYSGEN.
    Dynamic LAT DDBs are created whenever the maximum number of LAT
    ports exceeds the number of LAT ports defined in SYSGEN. Note that
    port 1 is never considered a Dynamic device, even if it is not
    defined.

Enter Port Number, or L (Dynamic LAT ports), or D (other Dynamic
Ports): 64

Enter Port Number, or L (Dynamic LAT ports), or D (other Dynamic
Ports): 65

Enter Port Number, or L (Dynamic LAT ports), or D (other Dynamic
Ports): RETURN

Current Tied Terminal Table List:

Index  Routine Reference  UCI,VGP  Partition Size  Ports Tied To Routine
-----  -----------------  -------  --------------  ---------------------
  1    Menu^PATIENT       PRD      48              5, 7, 11, 15
  2    ^FINANCIAL         PRD      48              64, 65

Enter Tied Terminal Index: RETURN

Select Option:
```

Port (Terminal) Definition

Use tThis option to create, display, or modify the parameters that define terminal devices to the system. The system prompts you to enter the information described below.

Port number

Enter the port number to be added to the configuration or the number of an existing port to edit its parameters. Or, enter an existing port number preceded by a minus sign to delete an entry. Port numbers can range from 1 through 19 (excluding 2), 64 through 199, and 256 and up. Port number 1 is assumed to be the main console device.

Port type

Every port that is defined to MSM has an associated type. MSM-UNIX port types include MSM (dedicated to MSM and cannot be used for login), UNIX (shared by MSM and UNIX), Parallel (for parallel printers), and LAT (for LAT terminal devices). The port type that is selected depends on the type of device to be accessed and its intended use.

Terminals that will be used to log on to MSM-UNIX are generally defined as UNIX shared devices. When a device is defined in this manner, the user logs on to UNIX and then enters the msm command to actually log on to MSM-UNIX. The MSM-UNIX logon can be avoided by using the tied terminal facility. UNIX shared devices cannot be OPENed from a user partition (they behave as if they are always in use, even when they are not in use by MSM).

Printer type devices can be defined as either MSM dedicated devices or pparallel devices. In either case, the device is locked by MSM when it is in use (it has been OPENed), and released back to UNIX when it is CLOSEd. MSM dedicated devices must be disabled for UNIX logon (there can be no getty assigned to the port). If it is necessary for a printer device to be shared by MSM-UNIX with other applications on the machine, it is more appropriate to use the Host System Spool device in MSM. Refer to “Using Peripheral Devices” in the *MSM User's Guide* for more information on printing through the UNIX Print Spooler.

Modems and other types of serial communications devices can be defined as logon devices or non-logon devices. If logon is to be allowed from the device, it must be defined as a UNIX shared device and must be enabled for logon in UNIX (a getty must be present for the device). Otherwise, it must be defined as an MSM dedicated device and must be disabled for logon in UNIX (no getty).

When the optional MSM-LAT feature is enabled in MSM-UNIX, a device type of LAT Port may be specified. The device type allows applications to create Host Initiated Connects (HIC connects) for terminals, printers, or modems attached to the system through LAT terminal servers. Refer to “MSM-LAT Services” in this manual for additional information on LAT devices.

Port address

Every communications port attached to the system has a unique device name assigned to it by UNIX. A device name (or device node as it is sometimes referred to in UNIX) is used to identify the information that describes the physical address and characteristics of the associated device. In MSM-UNIX, the port address is the UNIX device name (except for LAT ports when running in an Ultrix system; see below). The port address typically is in the form /dev/ttynnn, where nnn is a decimal or hexadecimal number.

Although some UNIX systems do not follow the normal naming conventions for identifying terminal ports, all UNIX system device names begin with the /dev/ prefix. We recommend that the UNIX console device (for example: /dev/console) be assigned a device number of 1 in MSM-UNIX. This is important since MSM system error messages usually are sent to this device. Refer to the UNIX documentation supplied with the system for additional information on terminal device names.

The port address can have special meaning when using LAT under the Ultrix operating system. When MSM-UNIX is installed under Ultrix, no support for LAT is actually implemented in MSM. Rather, MSM can access LAT ports through the DEC LAT services already available in Ultrix (assuming LAT is installed). In order for MSM to return a fixed value for the $IO special variable for LAT ports, MSM must know the LAT port name. MSM allows the LAT port name to be entered as the device address; it is entered in the form Port_number@Servername.

Tied terminal index

This option allows terminals to be automatically connected to specific application routines. An index in the Tied Terminal table indicating the name, location, and partition size of the application must have been previously defined. If no index is specified, the standard logon routine (%LOGON) supplied with the system is invoked. This option is ignored for single-user MSM systems.

Translate name

The translate name is an entry in the Device Translation tables which specifies input and output tables for translating 8-bit ASCII characters from different non-US character sets to a common set of characters.

Terminal line width

This option specifies the maximum number of characters which the terminal can display on a single line. If the next character to be displayed would force the $X Special Variable to exceed this number, the system automatically inserts a carriage return and a line feed character, and the character is displayed on the next line. Line sizes may range from 0 to 255 characters.

A value of 0 or 256 inhibits the automatic insertion of a carriage return and line feed sequence. The value of this option can be changed through the OPEN or USE commands. When escape sequences are used to control a device, the line width should be set to zero (0) so that a carriage return/line feed sequence is not inserted into the escape sequence.

Echo

This option controls whether or not MSM echoes characters back to the terminal as they are received. If automatic echo is desired, enter YES; otherwise, enter NO. The value of this option can be changed through the OPEN or USE commands.

Unless ECHO is turned off in SYSGEN, it is turned on immediately prior to each system prompt (the > programmer prompt). Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information.

Escape

This option controls how MSM interprets an ASCII Escape character when it is received. If escape processing is OFF, then the system treats the Escape character as if a carriage return had been entered, as long as Escape has not been explicitly removed from being a READ terminator.

If escape processing is ON, the system reads one or more additional characters from the terminal and then terminates the READ operation. The value of this option can be changed through the OPEN or USE commands. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information.

8-bit mode

This option controls how MSM treats ASCII characters received from the terminal. If 8-bit mode is OFF, the system treats all input characters as 7-bit ASCII and forces the high-order bit to zero. If 8-bit mode is ON, the system accepts full 8-bit ASCII characters. If the port is set to receive only 7-bit data (see "Data, Parity, and Stop Bits" below), then 8-bit characters cannot be received even if this option is on.

Care should be taken when using 8-bit mode since characters with an ASCII value greater than 127 are outside of the M ANSI Standard. If they are used as input to M commands and functions, they may produce unexpected results. This value can be changed through the OPEN or USE commands. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information.

Pass all

This option controls how MSM treats control characters received from the terminal. If pass-all mode is OFF, then the system interprets each control character according to the meaning assigned to it by MSM. If pass-all mode is ON, then no interpretation of control characters, other than the READ terminator (carriage return) is performed.

You also can disable the READ terminator by using device-specific options of the USE command, thus treating the carriage return as data. Refer to "Using the MSM System" in the *MSM User's Guide* for additional information on the control characters that are defined for MSM. The value of this option can be changed through the OPEN and USE commands. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information.

Line feed suppression

This option controls the output MSM generates when it encounters an exclamation point (!) as a format control in a READ or WRITE command. If line feed suppression is OFF, the system generates a carriage return and line feed sequence. If line feed suppression is ON, the system generates a carriage return only.

This feature is used with devices that generate their own carriage return upon line feed detection. The value of this option can be changed through the OPEN or USE commands. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information.

Modem control

This option defines how MSM handles modem devices. If this option is OFF, then the port is handled like a terminal device. In this case, the DTR signal is ignored by MSM, even though it may be required by the hardware. If it is required by the hardware, the signal must be supplied externally.

If the Modem Control option is ON, then bit 9 and bit 11 in *Opt5* of the OPEN and USE commands are honored. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information on modem control. This option cannot be changed through the OPEN and USE command parameters.

Uppercase setup

This option controls how MSM will handle lowercase characters found in a string that is input from a terminal. If uppercase setup is OFF, MSM accepts the lowercase characters without change. If uppercase setup is ON, MSM converts the characters to uppercase when they are received from the terminal. The value of this option can be changed through the OPEN or USE commands. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information.

Output only mode

This option controls what MSM does if a READ is directed to the terminal. If output only mode is OFF, normal READs are allowed from the terminal. If output only mode is ON, then a READ command that is directed to the terminal generates a <MODER> error.

Login prohibited

Use this option to specify to MSM whether or not the terminal device being defined is allowed to log on to the system. If this option is ON, the terminal is not allowed to log on to MSM. If this option is OFF, the terminal is allowed to log on to MSM. The value of this option cannot be changed through the OPEN and USE commands. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information.

CRT mode

Use this option to indicate to MSM whether or not the terminal being defined is a CRT-type device. If this option is ON, the device is known to be a CRT. If this option is OFF, the device type is not known by the system. The value of this option can be changed through the OPEN and USE commands. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information.

Device close option

By default when MSM opens an MSM device, it saves the current characteristics of the device, changes the characteristics as needed by MSM, and resets them when the device is closed. For UNIX operating systems that do not flush pending output properly, several Close Options are provided as a means of working around the deficiency on any system.

If the last few lines of printed reports are lost or garbled, then experiment with the various Close Options until you find one that works reliably.

Data, parity, and stop bits

This option is used to specify the Data bit, Parity, and Stop bit configuration that has been set up for the device. The system supports 7 or 8 data bits, even or odd parity, and 0, 1, or 2 stop bits. Any of the possible combinations of Data bits, Parity bits, and Stop bits can be specified. Refer to the hardware reference manuals supplied with the adapter boards or terminal device for information on the available options. The value of this option can be changed through the OPEN and USE commands. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information.

Baud rate

This option specifies the speed the terminal uses to transmit and receive information. Baud rates may range from 110 up to 38400, but baud rates above 9600 are not necessarily supported by all serial devices. Refer to the hardware reference manuals supplied with the system for additional information on available baud rates. The value of this option can be changed through the OPEN and USE commands. Refer to "Using Peripheral Devices" in the *MSM User's Guide* for additional information.

Sample terminal session

```
Select Option: 9 - Port (Terminal) Definition

Terminal Configuration:

Port    Port      Port         Tied   Line    Initial    Initial
Number  Type      Addr         Index  Width   Parms      Status
------  ----      -----        -----  -----   ------     -------
   1    UNIX      /dev/console          80               /Echo/CRT
   3    PARALLEL  /dev/tty2           132    Parallel    /OutDev
   4    MSM       /dev/tty6            80    1200,8B+1S  /CRT

Select Port Number: 5

Port Type:

    1 - MSM - MSM Dedicated Device
    2 - UNIX - UNIX Shared Device
    3 - PARALLEL Printer Port
    4 – LAT Port – connection to a specific port and server

Select Option: 1 - MSM Dedicated Device

Port Address: ?

    Enter the UNIX terminal device name in the form of /dev/name where
    name is from 1 to 14 characters.

Port Address: /dev/tty7
```

```
Tied Terminal Index <>: RETURN

Translate Name <>: ?

    Enter the translate name which defines special translation tables
    to be used for this device.  If no table is defined, the standard
    system tables will be used (no translation will be done for ASCII
    devices).
    Enter '-' to remove the current translate name.
    Enter '^L' for a list of currently defined translate names.

Translate Name <>: ^L

Table Name  Input  Output  Description
----------  -----  ------  -----------
VT220         3       4    Translate Table for VT220 crts
CONSOLE       1       2    Translate Table for System Console

Translate Name <>: VT220

Terminal Line Width <80>: ?

    Specify the maximum number of characters to be displayed per line
    of output after which MSM will output a Carriage Return and a Line
    Feed to automatically wrap lines around.

Terminal Line Width <80>: 132

Select Echo Option:

    1 - ON
    2 - OFF

Select Option <ON>: 2 - OFF

Escape Processing Mode:

    1 - OFF
    2 - ON

Select Option <OFF>: RETURN

8-bit Mode:

    1 - OFF
    2 - ON

Select Option <OFF>: RETURN

Pass All Mode:

    1 - OFF
    2 - ON

Select Option <OFF>: RETURN

Line Feed Suppression:

    1 - OFF
    2 - ON

Select Option <OFF>: RETURN

Modem Controlled:

    1 - NO
    2 - YES - AUTO ANSWER (DTR ON)
    3 - YES - NO AUTO ANSWER

Select Option <NO>: RETURN
```

```
Uppercase Setup:

    1 - OFF
    2 - ON

Select Option <OFF>: RETURN

Select 'Output-Only' Mode:

    1 - OFF (Input/Output Device)
    2 - ON (Output-only device)

Select Option <OFF>: 2 - ON (Output-only device)

Login Allowed:

    1 - YES
    2 - NO

Select Option <YES>: 2 - OFF

CRT Mode:

    1 - ON
    2 - OFF

Select Option <ON>: 2 – OFF

Select Device Close Option <1>: ?

    1 – Close and reset UNIX port characteristics
    2 – Do not reset the UNIX port characteristics
    3 – Do not close the port on the UNIX level
    4 – Wait 5 seconds during the close for the output to flush

Select Device Close Option <1>: RETURN

Select Initialization Option:

    1 - 7 Data Bits, No Parity, and 1 Stop Bit
    2 - 7 Data Bits, Even Parity, and 1 Stop Bit
    3 - 7 Data Bits, Odd Parity, and 1 Stop Bit
    4 - 7 Data Bits, No Parity, 2 Stop Bits
    5 - 7 Data Bits, Even Parity, and 2 Stop Bits
    6 - 7 Data Bits, Odd Parity, 2 Stop Bits
    7 - 8 Data Bits, No Parity, and 1 Stop Bit
    8 - 8 Data Bits, Even Parity, and 1 Stop Bit
    9 - 8 Data Bits, Odd Parity, and 1 Stop Bit
   10 - 8 Data Bits, No Parity, 2 Stop Bits
   11 - 8 Data Bits, Even Parity, and 2 Stop Bits
   12 - 8 Data Bits, Odd Parity, 2 Stop Bits

Select Option <7>: RETURN  7 - 8 Data Bits, No Parity, and 1 Stop Bit

Select Baud Rate:

    1 - 50
    2 - 75
    3 - 110
    4 - 134.5
    5 - 150
    6 - 200
    7 - 300
    8 - 600
    9 - 1200
   10 - 1800
   11 - 2400
   12 - 4800
   13 - 9600
   14 - 19200
   15 - 38400
```

```
Select Option <13>: 14 - 19200

Maximum number of devices (currently set to 10) <11>: ?

    Enter maximum number of devices (terminal, printers, modems).

Maximum number of devices (currently set to 10) <11>: RETURN

Terminal Configuration:

Port    Port     Port         Tied   Line    Initial    Initial
Number  Type     Addr         Index  Width   Parms      Status
------  ----     -----        -----  -----   ------     -------
   1    UNIX     /dev/console         80                /Echo/CRT
   3    PARALLEL /dev/tty2           132    Parallel    /OutDev
   4    MSM      /dev/tty6            80    1200,8B+1S  /CRT
   5    MSM      /dev/tty7            80    19200,8B+1S /OutDev
   Translate name='VT220'

Select Port Number: RETURN

Select Option:
```

External Calls Configuration

Use this option to automatically load an external call (XCALL) package into memory at system startup or to set up a default list of XCALL packages that are automatically loaded at system startup. Note that external calls are not supported on some MSM-UNIX implementations.

```
Select Option: 11 - External Calls Configuration

Select XCALL Option:

    1 - Files to load at startup
    2 - Default packages
```

Loading a package

The following terminal session illustrates how to add an XCALL package to the automatic load list. After the package is added to the list, it will be automatically loaded into memory each time the system is started.

```
Select Option: 1 - Files to load at startup

Enter Index Number: ?

    Enter an index number in the XCALL autoload table which will define
    a file to be loaded during system startup.

    Enter -index to delete an existing entry
    Enter ^L to list the current autoload table.

Enter Index Number: 1

File Name: ?

    Enter a file name which contains XCALL packages that will be loaded
    automatically during system startup.

File Name: /usr/lib/xcalls/mathpkg

Enter Index Number: ^L

Index   File Name
-----   ---------
  1     /usr/lib/xcalls/mathpkg

Enter Index Number: RETURN

Select XCALL Option: RETURN
```

Defining the Default Packages List

If a package name is not specified on the M statement that invokes an XCALL, MSM searches the Default Packages List (a list of XCALL packages). If no Default Packages List is defined, an internal package named MSM is treated as the only package in the list. If a list is defined, the MSM package is not automatically included in the list. To include it, you must add it to the list definition. After the Default Packages List is defined it is set up automatically in memory each time the system is started.

```
Select Option: 2 - Default packages

Enter Index Number: 1

Package Name: ?

    Enter a package name that will be added to the default packages
    list during system startup. If the internal package named MSM is to
    be included in the default list, it must be explicitly named in the
    list.

Package Name: MSM

Enter Index Number: 2

Package Name: MATH

Enter Index Number: ^L

Index   Package Name
-----   ------------
  1     MSM
  2     MATH

Enter Index Number: RETURN
```

Default Partition Size

Use this option to set the default partition size to be used for logon and jobs initiated with the JOB command. The partition size is specified as an integer value that indicates the maximum amount of memory that may be used by a job at any one time. When users logon to the system, they can specify the required partition size. Similarly, the required partition size may be specified as an operand on the JOB command. If the partition size is not specified at logon or on the JOB command, the default partition size is used.

The entire partition is not pre-allocated by the system. If the partition size is 64K but the job only requires 10K, then only 10K is assigned to the job. Additional memory within the partition is assigned as needed, until the maximum is reached.

```
Select Option: 12 - Default Partition Size

Default partition size: 48

New default partition size: ?

    Enter the partition size (in 1024 byte blocks) which the system
    will assign to each new job when it starts execution. This value
    will not be used if a partition size is explicitly specified via
    the JOB command, tied terminal table, or at sign-on time.

    The partition size must be at least 12K, and not more than 16384
    kilobytes (that is, 16 megabytes).
    It can be entered in any of the following formats.

    nnnk, nnnK, or nnn, sets the partition size to nnn*1024 bytes.
    nnm or nnM sets the partition size to nn**1024*1024 bytes,
    or nn megabytes.

Default partition size: 48

New default partition size: 64

Select Option:
```

Translation/Replication Table

The Translation/Replication Table option allows you to create, edit, list, and delete entries in the Translation and Replication tables, and to enable or disable the Translation function. MSM's Translation facility provides a mechanism whereby each reference to a global can be translated to reference a different global. The translated global name may be in a different UCI, Volume Group, or system.

Replication allows actual global updates (SETs and KILLs) to be duplicated in one or more UCIs. Unlike Translation, Replication maintains multiple copies of the same global in up to seven different UCIs, which can be on different Volume Groups and different systems. Global protection applies for Replication as it does for Translation.

Translation and Replication can be combined in such a way that a global or group of globals can be translated from one UCI to another while also being replicated elsewhere in the system. Optionally, you may specify a *master* UCI, which would maintain all of the LOCKs for the translated/replicated global(s).

The following example illustrates how to invoke the Translation/Replication Table option. When this option is selected, the SYSGEN internally invokes the TRANSLAT utility. For additional information on the TRANSLAT utility and a sample terminal session, refer to "Configuration Utilities" in the *MSM Utility Manual*.

```
Select Option: 13 - Translation/Replication Table

Select Translation/Replication Table Option:

    1 - Edit Translation Table
    2 - Enable Translation
    3 - Disable Translation
    4 - Edit Replication Table
    5 - Translation Table List
    6 - Replication Table List

Select Option:
```

Network Configuration

Use this option to define parameters for network system operation, define links, and define circuits for Manual Mode. When you select this option, the system displays the following sub-menu. Options 1-5 and Options 9-10 on the sub-menu are described in "MSM-NET Services" in this manual. Options 6-8 are described in "MSM-LAT Terminal Services" in this manual. Options 11-15 are described below.

```
Select Option: 15 - DDP And LAT Configuration

Select DDP Configuration Option:

     1 - DDP System Parameters
     2 - Link Definition
     3 - Circuit Definition
     4 - DDP Groups
     5 - Network Security
     6 - LAT System Parameters
     7 - LAT Groups
     8 - LAT HFS Access Control
     9 - OMI Translation Table
    10 - Display Network Configuration
    11 - Workstation Server Configuration
    12 - User-Defined Services
    14 – MSM-Activate Service Configuration
    15 - PDQweb Service Configuration

Select Option:
```

Workstation server configuration

Use this option to specify ports for connections between MSM-UNIX and MSM-Workstation.

```
Select Option: 11 - Workstation Server Configuration

Enter IP Port for Workstation TCP/IP Connect <33086>: ?

    Enter the IP port number on which the MSERVER utility will listen
    for connection attempts from MSM-Workstation.

    Enter '^' to exit this option without changing the value.

Enter IP Port for Workstation TCP/IP Connect <33086>: RETURN

Enter IP Port for Workstation SPX Connect <33080>: ?

    Enter the SPX port number on which the MSERVER utility will listen
    for connection attempts from MSM-Workstation.

    Enter '-' to clear the port number. If the number is cleared,
    connections will not be allowed for this protocol.
    Enter '^' to return to the previous prompt.

Enter IP Port for Workstation SPX Connect <33080>: 33080

Enter Password <>: ?

    Enter a password which MSM-Workstation must supply in order to
    connect to this server.

    Control characters and the semicolon (;) are not allowed as part of
    the password.

    Enter '-' to set the password to the null string.
    Enter '^' to return to the previous prompt.

Enter Password <>: ABC

Available Functions:
```

User-defined services

MSM supports user-defined services. A *service* is an application module that expects to receive a socket connection from a client system and then process application-dependent data from the client. MSM can automatically start services based on the parameters set up in SYSGEN. The following terminal session includes the online help for these parameters.

```
Select Option: 12 - User-Defined Services

Current User-Defined Services:

Index  Name     Protocol  Port  Password  Psize  Start UCI      Routine
-----  -------  --------  ----  --------  -----  ----- -------  --------
2      MAIL     TCPIP     2000            20     NO    MGR,MDC  ^MAILREC
2      VA Test  TCPIP     25              100    NO    MGR,MDC  ^zVATEST
```

```
Enter Service Index: ?

    User-Defined Services are M routines that are designed to process
    incoming data from a designated network socket.

    Select an index to this table:

    To create an entry - Specify a non-existing index (use '^L' for
    list of presently defined entries).

    To edit an entry - Specify the index of the desired entry.

    To delete an entry - Prefix the index of the entry to be deleted
    with a minus ('-'); e.g.: -5  will delete entry number 5.

    The ordering (value of the index) is not important. In fact, the
    entries are reordered alphabetically by name each time one is
    created or deleted.

Press <RETURN> to continue

Enter Service Index: 3

Enter Service Name: ?

    Enter a name to be associated with this service.
    This name will be displayed by the %SI utility when the service is
    active.

Enter Service Name: ABC

Enter Routine Reference: ?

Enter Service Index: 3

Enter Service Name: ?

    Enter a name to be associated with this service. This name will be
    displayed by the %SI utility when the service is active.

Enter Service Name: ABC

Enter Service Index: 3

Enter Service Name: ?

    Enter a name to be associated with this service. This name will be
    displayed by the %SI utility when the service is active.

Enter Service Name: ABC

Enter Routine Reference: ?

Select UCI: MGR

Enter Partition Size <256>: ?

    Enter a positive integer corresponding to the size (in 1K blocks)
    of the partition in which this routine will run. Partition sizes
    may range from 12K to 256K.

Enter Partition Size <256>: 256

Enter Password: ?

    Enter a password for this service. The password is not used by MSM,
    but is passed to the application in the variable %("PASSWORD"). It
    is up to the application to either check the validity of the
    password, or to ignore it.

Enter Password:
```

```
Enter TCPIP Port Number: ?

    Enter the IP port number to be associated with this service.
    Currently, only TCP/IP connections are supported. The MSERVER will
    listen on this port for a connection attempt from a client. Once a
    connection is established, the application will be called.

Enter TCPIP Port Number: 1000

Autostart?: ?

    Enter YES if this service is to be started automatically by MSM
    during the startup process.
    Enter NO if the service is to be manually started by the
    application software.

Autostart?: Y

Enter Service Index: ^L

Current User-Defined Services:

Index  Name     Protocol  Port  Password  Psize  Start UCI      Routine
-----  -------  --------  ----  --------  -----  ----- -------  --------
1      ABC      TCPIP     1000            256    YES   MGR,MDC  ^ABC
2      MAIL     TCPIP     2000            20     NO    MGR,MDC  ^MAILREC
2      VA Test  TCPIP     25              100    NO    MGR,MDC  ^zVATEST

Enter Service Index:
```

MSM-Activate service configuration

Use this option to configure the MSM-Activate Service. The following terminal session illustrates the configuration menu options. This menu includes different options depending on which features are enabled.

```
Select Option: 14 – MSM-Activate Service Configuration

    MSM-Activate Configuration Utility V4.4.0 Config 1

Select APIMGR Option:

    1 - Configure API Username/Passwords
    2 - Configure Security
    3 - Configure a different Server
    4 – Configure Server Ports

Select Option:
```

Configure MSM-Activate username and passwords

Use this option to set up usernames and passwords for the API Service. The prompt for MSM-Activate Version 3 password only occurs if Version 3 style connections are enabled using the Configure Security option.

```
Select Option: 1

Set MSM-Activate v3 Password

New v3 Password <NONE>: ?

    Enter the connection password. Enter <NONE> to disable password
    checking or <RETURN> to keep current selection.

Set MSM-Activate v4 Username/Passwords
```

```
Select USERNAME to edit: ?

    Enter the username whose properties you wish to edit,
    '^L' to list the current users, <RETURN> to return to the
    main menu, or '-<username>' to delete a user.

Select USERNAME to edit: ^L

Users currently set up

NONE

Select USERNAME to edit:
```

Configure security

Use this option to specify security parameters for the MSM-Activate Service.

```
Select Option: 2

Security options for server

Allow MSM-Activate Version 3 style connections (Y/N) <N>: ?

    Allow version 3 and version 4 clients to connect to the server, or
    just version 4 clients. Version 3 clients could present a security
    hole.

Allow MSM-Activate Version 3 style connections (Y/N) <N>: Y

Enter group whose properties you wish to edit: ?

    Enter the group you wish to edit; you set the security options for
    the group. Enter '^L' to list the current groups, <RETURN>
    or '^' to exit to main menu. If you wish to delete a group, enter
    '-<group name>'.

Enter group whose properties you wish to edit: ^L

Groups currently set up

DEFAULT

Enter group whose properties you wish to edit: DEFAULT

Allow group to Xecute commands <NO>: ?

    Allow the client to Xecute any command they wish on the server.
    This means that they can do anything a programmer could do. With
    the Xecute function disallowed, you can use the restrictions on the
    DO entry points to make the server more secure.

Allow group to Xecute commands <NO>: RETURN

Allow low level API commands <NO>: ?

    Allow the client to use the low-level MSM-Activate commands (only
    used by direct calls to the API DLL). They include things like read
    global, set local, increment lock, etc. If you are only using OLE
    with the API you can disable these functions to reduce security
    implications.

Allow low level API commands <NO>: RETURN

List DO patterns for group DEFAULT
----------------------------------

No   Pattern            UCI,SYS  Allow  Continue
1    .E1"^R".E          MGR,ABC  Y      N
```

```
Index number to change: ?

    Enter the number of the index you wish to edit/add. If you wish to
    delete an index, enter '-<index number>'. Enter <RETURN> to finish.

Index number to change: 1

Pattern to match <.E1"^R".E>: ?

    Enter pattern to match entry point with. Pattern match uses
    standard M syntax, entry point will look something like 'ENT^RT',
    so a pattern of '.E1"^R".E' will match all entry point in routines
    that begin with 'R'. This security feature enables you to restrict
    access to specific application routines.

Pattern to match <.E1"^R".E>: .E1"^PHARM".E

Uci,Sys to apply rule in <MGR,ABC>: ?

    Enter the name of the UCI,SYS you wish this pattern to apply to or
    '*' if you want it to apply to all UCIs. '^L' will list the UCI,SYS
    that are currently available.

Uci,Sys to apply rule in <MGR,ABC>: ^L

List of all the UCIs currently mounted

    MGR,ABC

Uci,Sys to apply rule in <MGR,ABC>: RETURN

If pattern matched allow Do <YES>: ?

    If the entry point matches the pattern then do we allow the entry
    point to be called? This means that some patterns can be ones to
    allow and some to disallow.

If pattern matched allow Do <YES>: RETURN

Continue searching for matches after this pattern <NO>: RETURN

List DO patterns for group DEFAULT
----------------------------------
No   Pattern            UCI,SYS  Allow  Continue
1    .E1"^PHARM".E      MGR,ABC  Y      N
```

Configure a different server

Use this option to switch to another server that is to be configured.

```
Select Option: 3

    Currently configuring server 1
    Switch to server (1-9) <1> ?

Enter a number between 1-9 or '^' to exit

    Currently configuring server 1
    Switch to server (1-9) <1> 2
```

Configure server ports

Use this option to specify the initial and final TCP ports to be used by the dedicated server jobs for the MSM-Activate service. For Version 3 connections, the client must switch TCP/IP port numbers after the initial login and before running any commands. This step is not required for Version 4 connections. Use this option to configure the range of TCP/IP port numbers that are used when the port switch takes place. If you only allow Version 4 connections, then you need not configure this option, which is removed from the list.

```
Select Option: 4

Configure Server Ports

Enter initial,final TCP port numbers <1667>: ?

    Enter initial and final TCP ports for the dedicated server jobs to
    use, separated by a comma. The final port may be omitted, allowing
    use of any port not less than the specified value.
    When specified, the final port may not precede the initial port.
    Enter <RETURN> to keep the current value.

Enter initial,final TCP port numbers <1667>:
```

PDQweb service configuration

Use this option to define the configuration for the MSM-PDQweb Service.

```
Select Option: 15 - PDQweb Service Configuration

Select UCI <MGR>: ?

    Enter the UCI and volume group in which PDQWeb is to be invoked. If
    the volume group is omitted, volume group 0 will be used. The UCI
    and volume group name must be separated by a comma.

    UCI and volume group names must be 3 uppercase letters.

Select UCI <MGR>: PRD,SYS

Enter Partition Size <256>: ?

    Enter a positive integer corresponding to the size (in Kilobytes)
    of the partition in which PDQWeb will run.

    Partition sizes may range from 32K to 16384K.

Enter Partition Size <256>: RETURN

Enter Password <>: ?

    Enter a password for PDQWeb.  This is used to validate the web
    server to make sure it is allowed to run M routines via PDQWeb.

    Enter '-' to delete a previously set password.
```

Tape Device Definition

Use this option to define industry-standard, 9-track magnetic tape devices and tape cartridge devices that are attached to the system. Refer to “Tape Devices” in this manual for additional information about magnetic tape drive support. Refer to “Using Peripheral Devices” in the *MSM User's Guide* for additional information on accessing tape devices.

In UNIX, each physical device can have one or more logical device names associated with it. Magnetic tape devices usually have a block device name (for example: /dev/mt/0), a character or raw device name (for example: /dev/rmt/0), and possibly other device names to control tape density, rewind options (for example: rewind tape when it is closed), and so on. Although there is no published standard for device names that set these options, most manufacturers follow similar conventions.

For example, with 9-track tape drives, a density other than the default density is often selected by appending a letter to the tape device name. Typically, the letter h indicates high density, m indicates medium density, and l indicates low density. However, the letter will often be interpreted differently depending upon the type of magnetic tape drive that is being accessed.

Assume that the letter h is used to designate the highest recording density that is available on the tape drive. It may indicate 6250 BPI on one brand of tape drive and may indicate 3200 BPI on another brand. The same is true for medium and low densities.

By default in most UNIX systems, when a 9-track tape or cartridge tape is closed, the system automatically rewinds the tape to the beginning. If the application does not want the tape to rewind when it is closed, then an alternate tape device name generally can be used to prevent it.

Typically, the letter n is appended to the device name to indicate that no rewind should be performed. For example, assume that a tape device name of /dev/rmt/0 causes the tape to rewind when it is closed. A device name of /dev/rmt/0n could be used to leave the tape positioned where it was after the last I/O operation, when a close is issued.

This information describes UNIX *conventions*, not UNIX standards. While many of the systems supported by MSM-UNIX follow these conventions, some do not. Refer to the UNIX documentation supplied with your system for additional information about the naming conventions for tape devices.

In general, you need not be concerned about the different tape names that set the various options. When configuring the tape device in MSM, use the base name (without the appended letters) and MSM will internally use the appropriate modified name, depending on the options selected in the OPEN and USE commands.

Because of limitations in certain UNIX operating systems, not all tape handling features of MSM-UNIX are available on all systems. Not all features available on 9-track tape drives are available on cartridge tape drives. Refer to “Tape Devices” in this manual for more information on defining tape devices and possible restrictions.

```
Select Option: 16 - Tape Device Definition

Device Number: ?

    Enter Tape Device # from 47 to 50.
    Enter -number to delete a tape device.

Device Number: 47

Raw Tape Device Name: ?

    Enter the UNIX raw tape device name. Do not include the density
    (h,m,l) or no-rewind-on-close (n) flags at the end of the device
    name.

Raw Tape Device Name: /dev/rmt/0

Cartridge Tape?: ?

    Enter YES if the device is a (1/4 inch) streaming cartridge tape.
    Enter NO if the device is a 9-track (1/2 inch) tape.

Cartridge Tape?: N

Device Number: 48

Raw Tape Device Name: /dev/rct0
```

```
Cartridge Tape?: Y

Current Tape Device Definitions are:

    Device     Device Name       Cartridge
    ------    --------------     ---------
    47      /dev/rmt/0            NO
    48      /dev/rct0             YES

Device Number: RETURN
```

Global Defaults

Use this option to modify certain default settings for newly created globals. In particular, the default collating sequence, default protection settings, and default journal status can be changed with this feature.

MSM maintains two default collating sequence flags, one for *inside the system*, and the other for *outside the system*. Inside the system means all globals created on the local MSM system. Outside the system refers to globals created on other systems using the Distributed Data Processing (DDP) option of MSM.

Default global protection is maintained separately for percent globals (globals whose names begin with the % sign) and for non-percent globals. Changing these defaults can allow application routines to create globals in others UCIs, included the Manager's UCI. Refer to "Globals" in the *MSM User's Guide* for additional information about global collating sequences and global protection attributes.

For each newly created global, select one of the following settings for global journaling:

- Globals can be set to always be journaled, regardless of whether the entire UCI is journaled.
- Globals can be set to never be journaled, also regardless of the UCI status.
- Globals can be set to honor the UCI status, being journaled only if the entire UCI is journaled.

```
Select Option: 17 - Global Defaults

Default Collating Sequence

  Inside the system..: NUMERIC
  Outside the system.: NUMERIC

Default Protection

% globals....:System = RWD  World = R     Group = R     User = RWD
non-% globals:System = RWD  World = NONE  Group = NONE  User = RWD

Set Default Collating Sequence

Default Collating Sequence inside the system <NUMERIC>: ?

    Each time a new global is created on this system, the collating
    sequence for the global will be set to this value (numeric or
    string).

Default Collating Sequence inside the system <NUMERIC>: RETURN

Default Collating Sequence outside the system <NUMERIC>: RETURN

Select Default Protection for % globals
```

```
Current Status ->System = RWD  World = R    Group = R   User = RWD

    1    N   - None
    2    R   - Read
    3    RW  - Read/Write
    4    RWD - Read/Write/Delete

Enter Protection for % globals > 4  RWD - Read/Write/Delete

Select Class for % globals

    1    System
    2    World
    3    Group
    4    User

Enter Class for % globals > 3 - Group

Select Default Protection for non-% globals

Current Status ->System = RWD World = NONE Group = NONE User = RWD

    1    N   - None
    2    R   - Read
    3    RW  - Read/Write
    4    RWD - Read/Write/Delete

Enter Protection for non-% globals > RETURN

Select Default Journaling status

Current Status -->     Journal only when UCI is journaled

    1    Always journal
    2    Journal only when UCI is journaled
    3    Never journal

Enter Journaling status > ?

    Select option 1 to journal newly created globals regardless of the
    journal status for the UCI.
    Select option 2 to journal newly created globals only when the UCI
    is journaled.
    Select option 3 to disable journaling for all newly created globals
    regardless of the journaling status for the UCI.

Enter Journaling status > 1

Select Option:
```

Lock Table Size

Use this option to change the maximum size of the Lock table. If a LOCK or ZALLOCATE command is issued which would expand the Lock table beyond this limit, the job will wait until memory is available (another job frees memory by unlocking a variable). If a single job owns all locks in the Lock table, then a <PGMOV> error occurs if it attempts to issue a lock that would overflow the table. The default Lock table size is 2-K bytes.

```
Select Option: 18 - LOCK Table Size

Maximum LOCK table size: 2K

New maximum LOCK table size: ?

    The system maintains a LOCK table that contains all of the LOCK and
    ZALLOCATE entries. The minimum LOCK Table size is 512 bytes; the
    maximum is 64K (65535 bytes). If a LOCK or ZALLOCATE command is
    executed that would cause the system to expand the LOCK Table
    beyond this limit, the job will wait until a lock is freed by
    another job. However, if a single job has filled the LOCK Table,
    then it will incur a <PGMOV> error.

Maximum LOCK table size: 2K

New maximum LOCK table size: 4

Select Option:
```

Display Configuration Parameters

This option displays all of the parameters that are specified for a particular configuration. Enter the name of the configuration to be displayed. If you do not enter a configuration name, MSM returns to the main option menu. MSM prompts you to enter an output device. The default device is the current terminal device.

```
Select Option: 19 - Display Configuration Parameters

Select Configuration <DFLT>: RETURN

Enter Output Device <1>: RETURN

------------------------------------------------------------------

Backspace, Line Delete Character

Current Backspace Character: Ctrl-H
Current Line Delete Character: Ctrl-X

------------------------------------------------------------------

Maximum Partitions

Maximum Partition Limit: 8

------------------------------------------------------------------

     .
     .
     .

     Additional Configuration Parameters

     .
     .
     .

Select Option:
```

Mode Flags

Use the Mode Flags option to enable or disable certain MSM features that are not ANSI-Standard. These features primarily facilitate conversion from other dialects of the M programming language and from other implementations of the ANSI Standard M language. These features are applied on a system-wide basis. To apply them to individual jobs, use the %MODESET utility.

The following sections describe these features. Refer to "Job Execution Utilities" in the *MSM Utility Manual* for information on the %MODESET utility.

```
Select Option: 20 - Mode Flags

Current Mode Settings:

Switch   Description                                          Status
------   ---------------------------------------------------  ------
0        Strict ANSI X11.1-1995 Conformance                   NO
1        Force $ZDATE to Always Return a Four Digit Year      NO
2        Set MSMSHELL programmer mode prompt to: [uci,vgp]>   NO
3        Version 3 Mode for Cross-UCI References in Functions NO
4        No Error Message for <ASYNC> Error                   NO
5        Wait for RVG Set or Kill Global request              NO
6        No Wait for DDP Set or Kill Global request           NO
7        Allow SET with Null Subscripts                       NO
8        Allow Reference to Null Subscripts                   NO
9        Version 2.0 Mode of Error Trapping                   NO
10       Always Set $T on JOB Command                         NO
11       Allow Routine Lines Greater than 255                 NO
13       Terminate ESC Processing on First Alphanumeric       YES
14       Treat Undefined Global Nodes as Null Valued          NO
15       Treat Undefined Local Nodes as Null Valued           NO

Enter switch to toggle > 11 Allow Routine Lines Greater than 255

Switch   Description                                          Status
------   ---------------------------------------------------  ------
0        Strict ANSI X11.1-1995 Conformance                   NO
1        Force $ZDATE to Always Return a Four Digit Year      NO
2        Set MSMSHELL programmer mode prompt to: [uci,vgp]>   NO
3        Version 3 Mode for Cross-UCI References in Functions NO
4        No Error Message for <ASYNC> Error                   NO
5        Wait for RVG Set or Kill Global request              NO
6        No Wait for DDP Set or Kill Global request           NO
7        Allow SET with Null Subscripts                       NO
8        Allow Reference to Null Subscripts                   NO
9        Version 2.0 Mode of Error Trapping                   NO
10       Always Set $T on JOB Command                         NO
11       Allow Routine Lines Greater than 255                 NO
13       Terminate ESC Processing on First Alphanumeric       YES
14       Treat Undefined Global Nodes as Null Valued          NO
15       Treat Undefined Local Nodes as Null Valued           NO

Enter switch to toggle > RETURN
```

Strict ANSI X11.1-1995 conformance

When this switch is set to Yes, the $NEXT function is not supported. The default setting for this switch is No.

Force $ZDATE to always return a four-digit year

When this switch is set to Yes, the $ZDATE function will always return a four-digit year. This feature can be useful in converting existing code to be year 2000-compliant. The default setting for this switch is No.

Set MSMSHELL programmer prompt to [UCI,VGP]

By default, the MSM shell program will prompt the user with the greater than sign (>). When this switch is set to Yes, the system will prompt with the UCI name and volume group name in addition to the greater than prompt ([MGR,SYS]>).

Version 3.0 mode for cross-UCI references in functions

Prior to Version 4.0, cross-UCI and cross-system references were made using square brackets ([ ]) around the UCI,VOL name. As of Version 4.0, this syntax is still supported, but functions such as $QUERY and $NAME, and the $ZREFERENCE system variable now return vertical bars in accordance with the ANSI Standard. This mode forces MSM to return brackets for applications that are coded to require them.

No error message for <ASYNC> error

By default, if a buffered DDP or Remote Volume Group transaction fails, an <ASYNC> error will be issued for the job that issued the SET or KILL. Enabling this mode flag suppresses the <ASYNC> error. The %DBSYNC utility can be used to explicitly check for errors.

Wait for RVG transaction

By default, when a SET or KILL is issued to a global on a Remote Volume Group which is locally mounted, the system will buffer the transaction, allowing the job to continue executing without waiting for a response. If the transaction does fail, an <ASYNC> error will occur unless Mode Flag #4 is turned on. Enabling this mode flag forces the system to wait for an acknowledgment for all RVG transactions.

No wait for DDP transaction

By default, when a SET or KILL is issued via DDP to a global on a remote Volume Group which is not locally mounted, the system will wait for a response from the remote system indicating success or failure before continuing. Enabling this mode flag allows the system to buffer the transaction and allow the job to continue executing without an acknowledgment from the remote system. If the transaction fails, an <ASYNC> error will occur unless Mode Flag #4 is turned on.

SET null subscripts

When this feature is enabled, global and local nodes can be created in which one or more subscript values are null. Globals and locals with null subscript values can also be killed. Be careful when using this option since it impacts the way the $ORDER function behaves. Note that $O(^ABC("")) skips the null node. Explicit $D(^ABC("")) syntax is needed to fully examine the global. Null subscripts collate differently when used in local or global variables. In locals, a null subscript collates after numeric subscripts and before string subscripts. In globals, null subscripts collate before all others.

Reference null subscripts

This function controls whether or not a program can reference (fetch) data from a global or local variable by specifying a key that contains one or more null subscripts.

Version 2.0 error trapping

In implementations of MSM prior to Version 2.1, whenever an error occurred, the system would automatically pop to the top of the execution stack before passing control to the error handling routine or returning to programmer mode. MSM now preserves the execution stack so that program execution can be continued after an error. This option allows the old style of error trapping to be the default mode.

SET $T on JOB command

The ANSI Standard indicates that after a JOB command is issued, the value of the $TEST Special Variable is changed only if the JOB command contained a timeout value. For compatibility with certain other mumps implementations, the option causes $T to be set after each JOB command regardless of whether or not it contains a timeout value.

Routine lines greater than 255

Normally, a line within a routine is limited to 255 characters. This option removes that limit. When this limit is removed, the only restriction is that a text line must fit entirely within one disk buffer and the compiled code for the text line must fit completely within one disk buffer. Since compiled code for a line of text varies depending on the content of the text, the maximum length of a text line cannot be precisely defined. In order to avoid <ISYNT> errors, the Full Screen Editor prohibits lines longer than 1,000 characters of text when this flag is on.

Terminate Escape Processing on first alphanumeric

Generally, when Escape Processing is enabled, MSM terminates a READ when the last character of the function key sequence is received. For example, the up-arrow key generates EscA, and MSM terminates the READ after the A is received. For VT200-type terminals, certain functions generate sequences with multiple numeric characters followed by the tilde (~) character. In this case, MSM terminates the READ when the tilde is received.

In Version 2.1, MSM terminated the READ when the first alphanumeric character was received. Although this prevents VT200-style function keys from being supported, certain applications might have been coded to work this way. Turning on this mode flag disables the processing of VT200-style function key sequences for backward compatibility with Version 2.1.

Treat undefined global nodes as null valued
Treat undefined local nodes as null valued

When a node within a global or local variable is referenced but does not exist, the system normally generates an <UNDEF> error. Use these options to indicate that the system should return a null value rather than treat the condition as an error. The application program then determines the action to be taken.

Edit Configuration Name/Comment

Use this option to rename a configuration and/or change the configuration comment (description). Once a configuration has been renamed, it can no longer be referenced by the old name. Enter the name of the configuration to be renamed and the new name to be associated with the configuration. MSM prompts you to enter a new configuration comment. If no comment is entered, the original comment remains unchanged.

```
Select Option: 4 - Edit Configuration Name/Comment

Select Configuration: DFLT

Enter New Configuration Name <DFLT>: PROD

Current Configuration Comment: DEFAULT CONFIGURATION

New Configuration Comment: ?

    Enter a comment to be associated with this configuration.

New Configuration Comment: Production Configuration

Select Option:
```

Delete Configuration

Use this option to delete an entire configuration. Once a configuration has been deleted, it cannot be recovered. If a configuration is inadvertently deleted, all of the parameters must be re-entered. Enter the name of the configuration to be deleted and confirm that the correct configuration has been identified. If no configuration name is supplied, the system returns to the main option menu.

```
Select Option: 5 - Delete Configuration

Enter name of configuration to delete: ?

   Enter the name of the configuration to be deleted.

   Enter '^L' to list the names of existing configurations.

Enter name of configuration to delete: ^L

Current Configuration: MDC

Available Configurations:

  Id: DFLT.............02/14/98: DEFAULT CONFIGURATION
  Id: MDC..............01/25/98: Production System
  Id: TST..............01/25/98: Test Configuration

Enter name of configuration to delete: TST

Are you sure <N>: YES ..deleted.

Enter name of configuration to delete: RETURN

Select Option:
```

Set Default Startup Configuration

Use this option to specify the name of the configuration to be used for system startup. If you do not supply the name of the configuration, the system returns to the main option menu.

```
Select Option: 6 - Set Default Startup Configuration

Select Configuration <DFLT>: ?

    Select a configuration to be edited (or created) by specifying its
    full name.  A configuration name may be any combination of letters
    and/or numbers with a maximum length of 14 characters.

    Enter '^L' to list the names of current configurations.

Select Configuration <DFLT>: MDC ..done.

Select Option:
```

UCI Management

Use this option to create, edit, delete, rename, and list the User Class Identifiers (UCIs) that are defined for the system. UCIs are defined for each Volume Group and are not tied to any particular configuration. Up to thirty (30) UCIs can be defined for each Volume Group, and they can span multiple volumes.

When creating a UCI, enter the Volume Group name, the name of the UCI (three uppercase characters), the location where the global and routine directory blocks are to be allocated, and the locations for routine and global expansion areas. The system also prompts for the upper limits on routine and global growth for the UCI. If no UCI management option is entered, the system returns to the main option menu.

The following example illustrates how the UCI Management option is invoked. When this option is selected, the SYSGEN internally invokes the UCIMGR utility. For additional information on the UCIMGR utility and a complete sample terminal session, refer to “Configuration Utilities” in the *MSM Utility Manual*.

```
Select Option: 7 - UCI Management

Select UCI Management Option:

    1 - Create New UCI
    2 - Rename Existing UCI
    3 - List UCI Status
    4 - Edit UCI Parameters
    5 - Delete Existing UCI
    6 - User Logon Identifier Edit
    7 – Edit UCI Routine Search Paths

Select Option:
```

System Configuration Parameters

The following system configuration parameters are configuration-independent settings that you can change to meet system requirements: Buffer Pool Size, Stack and Stap Size, Dynamic Caching Options, Maximum Number of Open Host Spool Files, and Number of Muserver Processes.

```
Select Option: 9 - System Configuration Parameters

Available Options:

    1 - Buffer Pool Size
    2 - Stack and Stap Size
    3 - Dynamic Caching Options
    4 – Maximum Number of Open Host Spool Files
    6 – Number of Muserver Processes

Select Option:
```

Buffer Pool Size

Use this option to specify the size of the disk buffer pool that is created by MSM. The *buffer pool* is used as a cache for all disk I/O operations. Whenever MSM requires a disk block, it searches the buffer pool to determine whether the block is already in memory. If it is, then the request is immediately satisfied. If it is not, then the requested block is read from disk.

Base the size of the buffer pool on the number of user partitions, the size of each partition, and the number of terminal devices that are defined. Generally, since all disk blocks (both routine and global blocks) are maintained in the cache, the larger the buffer pool, the better the performance.

When determining the size of the buffer pool, as a minimum each partition should have at least 64K and each terminal device at least .5K of cache. Refer to “Memory Management” in this manual for additional information on configuring the buffer pool, including requirements for UNIX shared memory.

This option does not actually set the size of the buffer pool; it sets the size of the shared memory area that MSM will get from UNIX. Within the shared memory area, MSM builds various system tables, which generally require approximately 300K of memory, plus an additional amount that depends on the other settings in SYSGEN. The remainder of the shared memory area is the buffer pool. Refer to “Configuring UNIX” in this manual for additional information on shared memory.

```
Select Option: 9 - Buffer Pool Size

Current Size of Buffer Pool: 1024

New Buffer Pool size: ?

    Enter the size of the system buffer pool.

    Value must be between 512k and 2047 megabytes inclusive.
    The format is nnn or nnnK (e.g., 512K) for a value in kilobytes,
    or nnnM (e.g., 9.5M) for a value in megabytes.

Current Size of Buffer Pool: 1024

New Buffer Pool size: 3M

Select Option:
```

Stack and Stap Size

When a partition is started (when a user logs on to the system or a JOB command is executed), MSM allocates two separate work areas, each of which is a fixed size. These areas are referred to as the stack and the stap. The *stack* is used to hold information associated with the nesting of program execution. Nesting occurs whenever a DO command, XECUTE command, or FOR command is encountered. The system saves all necessary information to restore the environment when a QUIT command is encountered.

The *stap* (or string accumulator) is used to store intermediate results of expression evaluation. It is also used as a buffer when a program is compiled as a result of a ZSAVE operation. The stap size controls the maximum size of the program that can be ZSAVEd and the complexity of the expressions that can be evaluated. Use this SYSGEN option to change the default sizes of the stap and stack.

```
Select Option: 2 - Stack and Stap Size

STACK size: 20K
STAP  size: 20K

New STACK size <20>: ?

    Enter the size (in K-bytes) of system stack space to be allocated
    to each job when it starts. This value affects how many levels of
    DO and EXECUTE commands can be nested. The value must be in the
    range 2 - 256.

New STACK size <20>: RETURN

New STAP size <20>: ?

    Enter the size (in K-bytes) of system stack space to be
    allocated to each job when it starts.  This value affects the
    maximum size of a routine that can be ZSAVED, and the maximum
    complexity of expressions that can be evaluated in an M command.
    The value must be in the range 2 - 256.

New STAP size <20>: 40
```

Dynamic Caching Options

The dynamic caching feature optimizes performance by saving compiled code in memory to avoid recompiling it each time the same code is encountered. XECUTE command arguments and indirection command arguments cannot be fully compiled until the time of execution. Therefore, such code is compiled as it executes. By saving the p-code from the compilation step, recompilation can be avoided if the same code is executed again by the same job. For applications that use the XECUTE command and/or argument indirection extensively, overall system performance can be slightly increased by caching these constructs.

```
Select Option: 3 - Dynamic Caching Options

Xecute Command Caching Option <DISABLE>: ?

    When Xecute Command Caching is enabled, P-Code resulting from
    compiling Xecute strings is retained and reused whenever possible.
    By reusing saved P-Code, the compilation  phase of the Xecute
    command is bypassed, resulting in a  performance gain.

    Enter ENABLE to enable caching, enter DISABLE to disable it.
    Enter '^' to return to the menu.

Xecute Command Caching Option <DISABLE>: ENABLE
```

```
Command Argument Indirection Caching Option <DISABLE>: ?

    When Command Argument Indirection Caching is enabled, P-Code
    resulting from compiling argument indirection strings is retained
    and reused whenever possible.  By reusing saved P-Code, the
    compilation phase of the indirection is bypassed, resulting in a
    performance gain.

    Enter ENABLE to enable caching, enter DISABLE to disable it.
    Enter '^' to go back to the previous option.

Command Argument Indirection Caching Option <DISABLE>: ENABLE
```

Maximum Number of Open Host Spool Files

Use this option to set the maximum number of UNIX lp spool files which can be concurrently open from MSM through device 55. The Host Spool Device (device 55) is a shared device that can be OPENed by more than one job. An internal table is created by MSM that manages all of the UNIX lp spool files which are being written to from MSM at any time. This option sets the size of this table. This table requires less than 1K of memory per 64 entries in the table.

```
Select Option: 5 - Maximum Number of Open Host Spool Files

Enter Number of Host Spool Files <16>: ?

    Enter the maximum number of Host Spool Files that can be
    concurrently open through device 55.

Enter Number of Host Spool Files <16>: 60
```

Number of muserver Processes

Use this option to set the number of muserver processes that MSM-UNIX will start up for UNIX file I/O. Each muserver process handles all UNIX file I/O for a particular job. UNIX file I/O includes I/O to Host File Server (devices 51-54), Tape (devices 47-50), Host Spool (device 55), TCP/IP Socket (device 56), and the JOB and JOBWAIT entry points of the %HOSTCMD utility. Since each muserver handles one job's requests for all of these I/O types, the number of active muserver processes is equal to the maximum number of MSM jobs that can concurrently be accessing any of these UNIX file I/O features.

```
Select Option: 5 - Number of Muserver Processes

Enter Number of Muserver Processes to Start <6>: ?

Enter the number of muserver processes to be created during system
startup.

    Each muserver process handles all host operating system I/O
    (Host File Server I/O, Tape I/O, Host Spooling I/O, TCP/IP
    Socket I/O, and the $$JOB and $$JOBWAIT subroutines of the
    %HOSTCMD utility) for a single job.  Therefore, the number of
    active muserver processes determines the maximum number of jobs
    that can be concurrently accessing the host operating system
    through any of these mechanisms.

Enter Number of Muserver Processes to Start <6>: 30
```

Database Definition

The MSM system supports a database structure that can include multiple independent databases called *volume groups*. MSM-UNIX supports up to 32 concurrently mounted volume groups, each of which may contain from one to eight separate volumes. A set of characteristics that is used to control a set of features is associated with each volume group: READ/WRITE access, LAT availability, type of p-code to generate, and maximum data length.

When a Volume Group is created, the system manager assigns it a three-character name (for example: SYS, TST, DEV). When a Volume Group is mounted, it is assigned an index number that indicates its relative position within the mount table (0 is the first mounted Volume Group, 1 is the second, and so on). The Volume Group from which MSM is booted is always assigned an index value of 0.

Within a volume group, each volume is assigned an index number that ranges from 0 through 7 that indicates the volume's relative position within the volume group. The first volume within a volume group is 0, the second is 1, and so on. Each volume also is assigned a volume label of up to eight characters. The volume label is designed to be a name that identifies the volume. For example, a volume label of SYSVOL3 could be assigned to volume 3 of volume group SYS.

A volume must be initialized before it can be used as part of a volume group. Initialization is performed by the installation process (for the first volume within the first volume group), the SYSGEN utility, or the DBMAINT utility supplied with the system. The initialization function creates the volume label block, clears all other blocks to zero, and creates the map blocks. If the volume group name assigned to the volume is not a currently mounted volume group, then the new volume becomes volume 0 of a new volume group. Otherwise, it is added as the next available index for the existing volume group.

When a new volume group is created, a UCI named MGR is created on it. In the MGR UCI, a ^SYS global is created, which is used to store the volume group characteristics. The MGR UCI can be renamed if desired, and used for application data. The ^SYS global should not be deleted, as this would prevent the DBMAINT utility from updating the volume group characteristics in the future.

The NOKILL flag is set for ^SYS to deter users from inadvertently deleting it. To be cautious, avoid using the MGR UCI for application purposes. For volume groups created under MSM Version 3.0 or earlier versions, a MGR UCI does not necessarily exist on all volume groups. If necessary, the DBMAINT utility creates a ^SYS global in the first UCI on any volume group (UCI #1) when the volume group characteristics are modified by the system manager.

The example below illustrates how to invoke the Database Definition option from the main SYSGEN menu. When you select this option, the SYSGEN utility internally invokes the DBMAINT utility. The BIJ and OLB utilities are available on this menu as Options 10 and 11 respectively. For information on DBMAINT and a terminal session, refer to "Configuration Utilities" in the *MSM Utility Manual*.

```
Select Option: 10 - Database Definition

Select Database Definition Option

    1 - Create a Database Volume
    2 - Reinitialize a Mounted Volume
    3 - Display Mounted Volume Groups
    4 - Mount a Volume Group
    5 - Unmount a Volume Group
    6 - Expand a Database Volume
    7 - Rename a Volume Group

Select Option:
```

Device Translation Tables

Different types of terminals commonly use different sets of ASCII characters to represent non-US characters. When this occurs, a standard method of storing these characters in the database must be designated so that data entered on one CRT is valid when displayed on another. The Device Translation tables are provided for this purpose.

Device Translation tables are user-coded and allow characters that differ between terminals to be stored as a single designated value. Each table is initialized by the utility to represent the complete 256-character extended ASCII character set.

Select the characters to be translated and enter the values to which they will be translated. A Translate Name is created which pairs two translation tables together, one designated for input, and the other for output. This Translate Name is entered when the terminal device is defined using the Port Definition option of the SYSGEN utility.

```
Select Option: 12 - Device Translation Tables

Select Device Translation Option:

    1 - Table Definition or Edit
    2 - Table List
    3 - Name Definition or Edit
    4 - Name List

Select Option:
```

Table Definition or Edit

Use this option to create a new input or output Translation table or to modify or delete an existing table. Either decimal or hexadecimal character values can be used. After you enter a modification, you can display the new table definition.

The following sample terminal session demonstrates both the table definition and editing functions. This example translates Ü on the console from ASCII 154 to ASCII 123 and Ö from 153 to 125. The left brace character ({) and the right brace character (}) are reserved and should not be used by the application program. The non-console terminal in the example uses the ASCII characters 162 and 164, respectively, for the same German characters. Tables 3 and 4 handle the translation for devices attached to the non-console terminals.

```
Select Option: 1 - Table Definition or Edit

Enter table number: ?

    Enter the number of a new input or output table to be created, or
    the number of an existing one to be edited.
    Enter '^L' for a list of existing tables.
    Enter -number to delete an existing table.

Enter table number: 1
A new table will be created.

Description: Input for Console

Enter 'D' for decimal values or 'H' for hex values: ?

    Enter 'D' if you will use decimal character values for
    modifications to the table.
    Enter 'H' if you will use hexadecimal character values for
    modifications to the table.
```

```
Enter 'D' for decimal values or 'H' for hex values: D

Enter a table modification: ?

    Enter a modification to the table. The format is value1=value2,
    where value1 is the numeric value of the character to be
    translated, and value2 is the replacement (translated) value. For
    example, an input table might specify 154=123 to translate the
    character $C(154) that is received from a terminal so that the M
    application actually receives $C(123). The same example would be
    specified as 9A=7B if the table is defined as using hexadecimal
    values.
    Enter -value1 to remove a modification.
    Enter '^L' to display the current table definition.

Enter a table modification: 154=123

Enter a table modification: 153=125

Enter a table modification: 155=25

Enter a table modification: ^L

Table #1, Description: Input for Console
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  153=125  154=123   155=25

Enter a table modification: -155

Enter a table modification: ^L

Table #1, Description: Input for Console
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  153=125  154=123

Enter a table modification: RETURN

Enter table number: 2
A new table will be created.

Description: Output for Console

Enter 'D' for decimal values or 'H' for hex values: D

Enter a table modification: 123=154

Enter a table modification: 125=153

Enter a table modification: RETURN

Enter table number: 3
A new table will be created.

Description: Input for VT220 crts

Enter 'D' for decimal values or 'H' for hex values: D

Enter a table modification: 162=123

Enter a table modification: 164=125

Enter a table modification: RETURN

Enter table number: 4
A new table will be created.

Description: Output for VT220 crts

Enter 'D' for decimal values or 'H' for hex values: D
```

```
Enter a table modification: 123=162

Enter a table modification: 125=164

Enter a table modification: RETURN

Enter table number: ^l

Device Translation Table Display

Table #1, Description: Input for Console
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  153=125  154=123

Table #2, Description: Output for Console
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  123=154  125=153

Table #3, Description: Input for VT220 crts
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  162=123  164=125

Table #4, Description: Output for VT220 crts
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  123=162  125=164

Enter table number: RETURN
```

Table List

Use this option to display information about defined Translation table(s). The list includes a description of the table function (whether used for input or output, console or terminals), the type of translation (for example: from ASCII to ASCII) whether values are formatted as decimals or hexadecimals, and the specific characters that are being modified to support the translation. A sample table list is provided below.

```
Select Option: 2 - Table List

Device Translation Table Display

Table #1, Description: Input for Console
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  153=125  154=123

Table #2, Description: Output for Console
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  123=154  125=153

Table #3, Description: Input for VT220 crts
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  162=123  164=125

Table #4, Description: Output for VT220 crts
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  123=162  125=164
```

Name Definition or Edit

Use this option to define or modify the translate name that is used to link two translation tables, one for input and one for output. You also may list or delete existing translate names.

The following terminal session demonstrates creation of a new translate name. After you enter the new name, the system prompts you to specify the table numbers of the translation tables that are to be associated for input and output.

```
Select Option: 3 - Name Definition or Edit

Enter translate name: ?

    Enter the translate name to be created, or an existing one to be
    edited.  Up to 8 alpha-numeric characters are allowed.
    Enter '^L' for a list of existing translate names.
    Enter -name to delete an existing name.

Enter translate name: CONSOLE
Creating a new translation name.

Input table: ?

    Enter the number of an existing translation table that will be used
    for input translation for this name.
    Enter '^L' for a list of tables currently defined.

Input table: ^L

Device Translation Table Display

Table #1, Description: Input for Console
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  153=125  154=123

Table #2, Description: Output for Console
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  123=154  125=153

Table #3, Description: Input for VT220 crts
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  162=123  164=125

Table #4, Description: Output for VT220 crts
Translation type: ASCII -> ASCII      Values are in decimal
Modifications:  123=162  125=164

Input table: 1

Output table: 2

Description: Translate Table for Console

Enter translate name: VT220
Creating a new translation name.

Input table: 3

Output table: 4

Description: Translate Table for VT220 crts

Enter translate name: RETURN
```

Name List

Use this option to list existing translate names and their descriptions, as well as the numbers of Translation tables that are paired for input and output. A sample name list is provided below.

```
Select Option: 4 - Name List

Device Translation Name Display

Table Name  Input   Output   Description
----------  -----   ------   -----------
CONSOLE       1        2     Translate Table for Console
VT220         3        4     Translate Table for VT220 crts
```

Mnemonic Namespaces

Use the Mnemonic Namespaces feature of MSM to develop M applications that are relatively device-independent in handling low-level device functions (for example: clearing a terminal screen or backspacing one block on a tape). You can use Mnemonic Namespaces to access many of the device types supported by MSM.

The following device types can be supported through Namespaces: Terminal devices, Sequential Block Processor (SBP) devices, Host File Server (HFS) devices, Interjob Communication (IJC) devices, Magnetic Tape devices, the MSM Spool Device, and the Host System Spool Device. Refer to "Using Peripheral Devices" in the *MSM User's Guide* and to "MSM Commands" (the OPEN, USE, READ, and WRITE commands) in the *MSM Language Reference Manual* for additional information on Mnemonic Namespaces.

MSM supports two types of Mnemonic Namespaces: built-in Namespaces and user-defined Namespaces. The standard distribution for MSM-UNIX includes two built-in Namespaces and one user-defined Namespace. The built-in Namespace X3.64-1979, is commonly referred to as the ANSI terminal Namespace. The built-in Namespace ZWINTERM allows windowing capabilities on dumb terminals. Refer to "Mnemonic Namespaces" in the *MSM Language Reference Manual* for complete descriptions of all mnemonics defined for these two Namespaces.

The user-defined Namespace X3.64 TEMPLATE is distributed in a host file called ANSI.NAM. Use this Namespace as a template for creating other namespaces. To import the X3.64 TEMPLATE Namespace into the MSM database, use the Import a Namespace option, then copy the Namespace to a new name and edit it as needed. This Namespace provides the complete set of ANSI mnemonics.

Because ANSI terminals are the most frequently used type of terminal in M applications, the Namespace for this terminal type is built directly into the system. Because a built-in Namespace is coded directly into the MSM system monitor rather than through user-supplied M code, overhead is significantly reduced when using this Namespace.

Since the X3.64-1979 Namespace is built-in, you cannot edit or delete it, and the mnemonics defined within the Namespace cannot be listed through this option of the SYSGEN utility. Refer to "Mnemonic Namespaces" in the *MSM Language Reference Manual* for a complete list of the mnemonics associated with the X3.64-1979 Namespace and the actions they perform.

Use the Mnemonic Namespace option of the SYSGEN utility to create, edit, and delete user-defined Namespaces. To create a Mnemonic Namespace, you must first assign it a unique name (an identifier). The name can be from one to 15 characters in length and may include any alpha, numeric, or special characters.

After a Namespace is created, define one or more Mnemonics within it. A *mnemonic* is a shorthand name associated with a specific device function. When a mnemonic is created, you must specify its name, which can be from one to 15 characters in length. The name must begin with an alpha character or ? (question mark). The remaining characters can be any valid alpha or numeric characters.

Next, indicate whether or not the mnemonic function will affect the values of the $X and $Y special variables. If you indicate that the values are not updated by the function, the system ensures that $X and $Y contain the same values at the end of the function as they did at the start. If you indicate that the values are affected, you must supply the code necessary to ensure that $X and $Y are updated properly.

You must next indicate the number of parameters that are required by the function. Specify a value from 0 through 9 for the minimum and maximum number of parameters that can be supplied for the function. You can specify the same or different values for both parameters. If you specify 0 for both the minimum and maximum number of parameters, no parameters will be allowed when the mnemonic is invoked.

Next, supply the actual M code necessary to perform the device-handling function associated with the mnemonic. Within this code, you can obtain the parameters that were passed to the mnemonic by referencing a set of special variables maintained by MSM for mnemonic functions. The special variables $1 through $9 contain the parameters that were passed to the function. The $0 special variable contains a count of the number of parameters that were passed by the function.

In the specified M code, you can access the $0 through $9 special variables in the same manner that standard MSM special variables (for example: $X, $Y, $ZA) can be accessed. The M code associated with the mnemonic function cannot alter the value of these special variables.

The following example code illustrates how an M program can use a mnemonic to position the cursor on the terminal screen. Assume that a user-defined Mnemonic Namespace of Q102 was previously defined and within this Namespace, a mnemonic of CUP was defined for cursor positioning. The following code positions the cursor to row 10, column 55 on the screen before writing the specified text:

```
USE 0::"Q102"
WRITE /CUP(10,55),"Customer Number:"
```

In this example, two parameters are passed to the CUP mnemonic. The first parameter (10) is contained in the $1 variable, and the second parameter (55) is contained in the $2 variable. The $0 variable contains 2 to indicate that two parameters were passed. The M code associated with the CUP mnemonic is shown below:

```
W *27,$C($1+31),$C($2+31) S $Y=$1,$X=$2
```

The WRITE command in this example outputs the actual escape sequence required by the terminal to position the cursor. The SET command updates the values of $X and $Y to match the new cursor position. Refer to “MSM Special Variables” in the *MSM Language Reference Manual* for additional information on the $X and $Y special variables.

The following terminal session illustrates how to invoke the Mnemonic Namespaces option. Detailed examples that illustrate how to create, edit, and delete Namespaces are provided below.

```
Select Option: 13 - Mnemonic Namespaces

Select Mnemonic Namespace Option:

    1 - List All Defined Namespaces
    2 - Add New Namespace
    3 - Copy Namespace To New Name
    4 - Edit Existing Namespace
    5 - Delete Existing Namespace
    6 - Rename Existing Namespace

Select Option:
```

List All Defined Namespaces

Use this option to list all of the Mnemonic Namespaces that are known to the system. This option lists the names of the built-in Namespaces and the names and definitions of user-defined Namespaces.

Enter the output device where the listing is to take place. The following sample terminal session illustrates the List function.

```
Select Option: 1 - List All Defined Namespaces

Enter output device <76>: RETURN

Namespace names defined internally to MSM-UNIX:
          `X3.64-1979'
          `ZWINTERM'

Namespace names defined by SYSGEN:
          None.

Select Option:
```

Create New Namespace

Use this option to create a new Mnemonic Namespace. Enter a unique identifier to be associated with the Namespace, one or more mnemonic names, and the complete definition for the mnemonic. The following terminal session illustrates this function.

```
Select Option: 2 - Add New Namespace

Enter new Namespace name: ?

    Enter the name of a Mnemonic Namespace to be defined. The name can
    consist of up to 15 alpha numeric characters.
    Enter '^L' for a list of currently defined Namespaces.

Enter new Namespace name: QUME

Enter mnemonic name: ?

    Enter name of mnemonic to be edited, added, or deleted.
    To delete a mnemonic, prefix it with a `-'.
    Enter `^L' for a display of defined mnemonics.
    Mnemonic name format is:
        1 to 15 characters,
        Leading character is limited to a letter or `?'
        Remaining characters may be a letter or numeric
    Enter `^' to exit.

Enter mnemonic name: CLEAR
```

```
Does mnemonic change $X <N>: ?

    Respond with Y if the mnemonic causes $X to change. If you answer
    yes, make sure the M code you enter for the mnemonic does a `SET
    $X=value' after completing.
    Respond with N if the mnemonic does not cause $X to change. If you
    answer no, the value of $X will be preserved during execution of
    the mnemonic.
    Respond with `^' to backup to the previous prompt.

Does mnemonic change $X <N>: N

Does mnemonic change $Y <N>: ?

    Respond with Y if the mnemonic causes $Y to change. If you answer
    yes, make sure the M code you enter for the mnemonic does a `SET
    $Y=value' after completing.
    Respond with N if the mnemonic does not cause $Y to change. If you
    answer no, the value of $Y will be preserved during execution of
    the mnemonic.
    Respond with `^' to back up to the previous prompt.

Does mnemonic change $Y <N>: N

Enter minimum number of parameters <0>: ?

    Enter the minimum number of parameters required for successful
    execution of the mnemonic.
    The minimum acceptable value is 0.

Enter minimum number of parameters <0>: 0
Enter maximum number of parameters <0>: ?

    Enter the maximum number of parameters required for successful
    execution of the mnemonic.
    The minimum acceptable value is 0.

Enter maximum number of parameters <0>: 0
Enter M code for performing mnemonic function > ?

    Enter M code to perform the device-specific function associated
    with the mnemonic.  For example, the following code will move the
    cursor to row $1, and column $2 on an ANSI terminal.  The $1
    special variable contains the value of the first argument of the
    mnemonic, and the $2 variable contains the value of the second
    argument. Note that the code to update $X and $Y, if needed must be
    specified here as well.

    W *27,"[",$1,";",$2,"H" S $Y=$1,$X=$2

Enter MUMPS code for performing mnemonic function > W *27,"Y"

Mnemonic          Set    Parms
  name           $X/$Y  Min/Max  MMPS Code
---------------  -----  -------  ---------
CLEAR             ---     0,0    W *27,"Y"

     .
     .
     .

     Additional Mnemonic Definitions

     .
     .
     .

Enter mnemonic name: CUP
Does mnemonic change $X <N>: Y
Does mnemonic change $Y <N>: Y
Enter minimum number of parameters <0>: 2
```

```
Enter maximum number of parameters <2>: 2
Enter MUMPS code for performing mnemonic function
    > W *27,$C($1+31),$C($2+31) S $Y=$1,$X=$2

Mnemonic          Set    Parms
  name           $X/$Y  Min/Max  M Code
---------------  -----  -------  -------------------------
CUP              $X,$Y    2,2    W *27,$C($1+31),$C($2+31)
                                    S $Y=$1,$X=$2
Enter mnemonic name: ^L

Mnemonic          Set    Parms
  name           $X/$Y  Min/Max  M Code
---------------  -----  -------  -------------------------
BLINK             ---     0,0    W *27,"G2"
CLEAR             ---     0,0    W *27,"Y"
CUP              $X,$Y    2,2    W *27,$C($1+31),$C($2+31)
                                    S $Y=$1,$X=$2
REVERSE           ---     0,0    W *27,"G4"
UNDERLINE         ---     0,0    W *27,"G8"

Enter mnemonic name: RETURN

Select Option:
```

Copy Namespace to New Name

Use this option to copy an existing Mnemonic Namespace to a new Mnemonic Namespace (create a new Namespace which is identical to an old Namespace). Generally, this option is used when the new Namespace is very similar to the old one. After the new Namespace is created, you can edit it.

Enter the name of an existing Namespace and a unique identifier to be associated with the new Namespace. The following sample terminal session illustrates the Copy function.

```
Select Option: 3 - Copy Namespace to New Name

Select Namespace to be copied: ?

    Enter the name of the Mnemonic Namespace to be copied. The name you
    choose must be defined, but cannot be a built-in Namespace.
    Enter `^L' for list of currently defined Namespaces.

Select Namespace to be copied: ^L

Namespace names defined internally to MSM-UNIX:
          `X3.64-1979'
          `ZWINTERM'

Namespace names defined by SYSGEN:
          `Q102'

Select Namespace to be copied: Q102

Enter new Namespace name: Q105

.....! done.

Select Option:
```

Edit Existing Namespace

Use this option to edit one or more mnemonic definitions within an existing Mnemonic Namespace or to add new mnemonic definitions to an existing Mnemonic Namespace.

Enter the name of an existing Namespace and a mnemonic identifier. If the identifier exists, the system prompts you to enter editing information. If it does not exist, enter the information necessary to create the entry. The following sample terminal session illustrates the Edit function.

```
Select Option: 4 - Edit Existing Namespace

Select Namespace to edit: ?

    Enter the name of an existing Namespace to edit.  You will be
    allowed to create new mnemonics for this Namespace, or edit or
    delete existing ones.
    Enter `^L' for list of currently defined Namespaces.

Select Namespace to edit: ^L

Namespaces defined internally to MSM-UNIX:
          `X3.64-1979'
          `ZWINTERM'

Namespaces names defined by SYSGEN:
          `QUME'

Select Namespace to edit: QUME

Enter mnemonic name: ?

    Enter name of mnemonic to be edited, added, or deleted.
        To delete a mnemonic, prefix it with a `-'.
        Enter `^L' for a display of defined mnemonics.
        Mnemonic name format is:
            1 to 15 characters,
            Leading character is limited to a letter or `?'
            Remaining characters may be a letter or numeric
        Enter `^' to exit.

    Enter mnemonic name: ^L

Mnemonic          Set    Parms
  name           $X/$Y  Min/Max  M Code
---------------  -----  -------  -------------------------
BLINK             ---     0,0    W *27,"G2"
CLEAR             ---     0,0    W *27,"Y"
CUP              $X,$Y    2,2    W *27,$C($1+31),$C($2+31)
                                   S $Y=$1,$X=$2
REVERSE           ---     0,0    W *27,"G4"
UNDERLINE         ---     0,0    W *27,"G8"

Enter mnemonic name: EL
Does mnemonic change $X <N>: N
Does mnemonic change $Y <N>: N
Enter minimum number of parameters <0>: 0
Enter maximum number of parameters <0>: 0
Enter MUMPS code for performing mnemonic function > W *27,"[G1"

Mnemonic          Set    Parms
  name           $X/$Y  Min/Max  M Code
---------------  -----  -------  ---------------------------
EL                ---     0,0    W *27,"[G1"

Enter mnemonic name: UNDERLINE
Does mnemonic change $X <N>: N
```

```
Does mnemonic change $Y <N>: N
Enter minimum number of parameters <0>: 0
Enter maximum number of parameters <0>: 0

Edit M code for mnemonic:
    > W *27,"G8"
      Replace: G8  With: G9  Replace: RETURN
    > W *27,"G9"
      Replace: RETURN

Mnemonic          Set    Parms
  name           $X/$Y  Min/Max  M Code
---------------  -----  -------  ----------------------------
UNDERLINE         ---     0,0    W *27,"G9"

Enter mnemonic name: RETURN

Select Option:
```

Delete Existing Namespace

Use this option to delete a user-defined Namespace. The system does not allow built-in Namespaces to be deleted.

Enter the name of the Namespace to be deleted. The following sample terminal session illustrates the Delete function.

```
Select Option: 5 - Delete Existing Namespace

Select Namespace to be deleted: ?

    Enter the name of the Mnemonic Namespace to be deleted.
    Enter `^L' for list of currently defined Namespaces.

Select Namespace to be deleted: ^L

Namespace names defined internally to MSM-UNIX:
          `X3.64-1979'
          `ZWINTERM'

Namespace names defined by SYSGEN:
          `Q102'
          `Q105'

Select Namespace to be deleted: Q105

Are you sure <N>: Y

Deleting........done.

Select Namespace to be deleted: RETURN

Select Option:
```

Rename Namespace

Use this option to rename a user-defined Namespace. The system does not allow built-in Namespaces to be renamed.

At the appropriate prompts, enter the old and new names of the Namespace. The following sample terminal session illustrates the Rename function.

```
Select Option: 6 - Rename Existing Namespace

Select Namespace to be renamed: ?

    Enter the name of the Mnemonic Namespace to be renamed.
    Enter `^L' for list of currently defined Namespaces.

Select Namespace to be renamed: ^L

Namespace names defined internally to MSM-UNIX:
          `X3.64-1979'
          `ZWINTERM'

Namespace names defined by SYSGEN:
          `Q102'
          `Q105'

Select Namespace to be renamed: Q102

Enter new Namespace name for `Q102': QUME

.....! done.

Select Option:
```

Export Namespaces

Use this option to save a user-defined Namespace to an external file so that it can be copied to another system. Select an output device and the Namespaces to be exported. The following sample terminal session illustrates the Export function.

```
Select Option: 7 - Export Namespaces

Enter output device <HFS>: RETURN

File Name: QUME.NAM

Enter size of save medium (if applicable): RETURN

Enter comment for dump header: Namespaces for QUME crts

Mnemonic namespace selector: Q102

Mnemonic namespace selector: Q105

Saving ...

^SYS
^SYS

Save complete.
```

Import Namespaces

Use this option to load user-defined Namespaces from an external file that was created by the Export Namespaces option. At the system prompt, enter the input device. The following sample terminal session illustrates the Import function.

```
Select Option: 8 - Import Namespaces

Enter input device <HFS>: RETURN
File Name: ANSI.NAM
Mnemonic namespace(s) saved at  1:20 PM  27-JUL-97.
Header comment is: ANSI X3.64 Template mnemonic namespace

Selective restore <N>: ?

    If you chose selective restore, you will be prompted for each
    Mnemonic Namespace. You must then respond to each prompt to restore
    the Namespace or skip to the next one.
    Enter NO to restore all Mnemonic Namespaces that were saved.
    Enter ^ to exit without restoring any Mnemonic Namespaces.

Selective restore <N>: RETURN
Restoring...
Mnemonic Namespace: X3.64 TEMPLATE     ... Restored
Restore Complete
```

Journaling Management

Use this option to configure and operate all global journaling features using the JRNL utility. The following example illustrates how to invoke the Journaling Management option.

Refer to “Resilient Systems” in this manual for a complete description of the options displayed on the Journaling Management menu.

```
Select Option: 14 - Journaling Management

Journaling not active

Select Journal Option:

    1 - Activate (Start) Journaling
    2 - Deactivate (Stop) Journaling
    3 - Show (Display) Journal Spaces
    4 - Configure Journaling
    5 - Mark Journal Space as Reusable
    6 - Switch to New Journal Space
    7 - Restore (Dejournal) From Journal Space
    8 - Print a Journal
    9 - Cross-System Journaling Options

Select Option:
```

MSM System Operation

Overview

This chapter describes the procedures for starting up the system, shutting down the system, and backing up the database.

System Startup

To start up MSM, first ensure that the UNIX system is fully initialized and is operating in multi-user mode. Continue by logging on to the root userid of UNIX or to any other userid that has root privileges. If you are not sure how to do this, refer to the UNIX documentation supplied by the manufacturer.

After logging on to UNIX, use the change directory command (cd) to switch to the directory that contains the MSM system modules (the directory into which the distribution system was loaded using the tar command). Once you are in the proper directory, enter the msmstart command (in either lowercase or uppercase) using the following format:

./msmstart {-b buffers} {-d database} {-g config}

{-i} {-k stack} {-p partitions}

{-t terminals} {-v wait} {-w stap}

{/{no}autoconfig{=config}} {/key}

The parameters enclosed in braces ({parm}) are optional and are used to override the standard default startup parameters. The options specified on the MSM startup command also override the parameters contained in the CONFIG.MSM file. For information on the CONFIG.MSM file, refer to “Installing the MSM System” in this manual.

Do not include the braces around the parameters ({ }) when you specify any of the optional parameters. Each option is separated from the previous option by one or more space characters.

-b buffers

Use this parameter to specify the size of the disk buffer pool. When this parameter is specified, it overrides the value in the CONFIG.msm file. Alternatively, you can modify the value in the CONFIG.msm file by changing the buffer pool parameter with the SYSGEN utility.

-d database

Use this parameter to specify the name of the UNIX file that contains the MSM database. The name specified must be the fully qualified file name (directory path and file name) for volume zero of the System volume group of the database to be mounted during system startup.

If this parameter is omitted, the system uses the name that is stored in the CONFIG.msm file. This file is created during installation of the MSM system. The installation process creates a database named *database.msm* on the drive and in the directory you specify. This fully-qualified name is stored in the CONFIG.msm file. For additional information on the CONFIG.msm file, refer to "Installing the MSM System" in this manual.

-g config

Use this parameter to specify the name of the configuration file that is to be used for initialization. This file contains the host file name of the volume group that is to be mounted, the number of partitions, the buffer cache size, and so on. Refer to "Installing the MSM System" in this manual for additional information.

If this parameter is omitted, then the CONFIG.msm file is assumed to contain the startup options.

-i

This flag causes MSM to bypass the automatic execution of the system startup routine STU and place the user in Baseline Mode. In Baseline Mode, only the System Manager's UCI is recognized, and only the first volume of the startup volume group is mounted. If this option is omitted, then STU is automatically invoked to initialize the system based on the parameters set by the SYSGEN utility.

This option can be used to trouble-shoot problems with the startup mechanism or to change SYSGEN settings that are causing the normal startup to fail. Once the changes have been made, shut down MSM using the SSD utility and then restart it normally. For additional information on the system generation procedure, refer to "Generating the System" in this manual.

-k stack

Use this parameter to specify the size of the system stack, which is used for DO, FOR, XECUTE, and Indirection. If the value of this parameter is too small, <STKOV> errors will occur. The size is specified in units of 1-K bytes.

If this parameter is omitted, the stack size is taken from the CONFIG.msm file.

-t terminals

Use this parameter to specify the maximum number of devices (terminals, printers, modems, and so on) that can be connected to MSM. From this information, the system dynamically creates the terminal Device Descriptor Blocks (DDBs) that are used to control the devices. MSM also creates the input and output ring buffers (256 bytes for input and 512 bytes for output) associated with the devices.

-v {seconds}

Use this parameter to abort the startup. It is useful when msmstart is part of an extended system startup script that automatically starts various application and/or system software. Enter Y to abort the startup or N to continue the startup without waiting any longer. If you do not enter a response at the prompt, then the MSM startup continues after waiting for a specified number of seconds (or 10 seconds if no time is specified).

-w stap

Use this parameter to specify the size of the stap (string accumulator), which is used for concatenates, string expressions, XECUTEs, and compiling programs. If the value of this parameter is too small, <STKOV> errors occur when large programs are compiled, especially if ZLOAD and ZSAVE commands are performed within an XECUTE statement. The size is specified in units of 1-K bytes.

If this parameter is omitted, the stap size is taken from the CONFIG.msm file.

/autoconfig{=config}

Use this parameter to indicate whether or not the system should prompt the system manager for the name of a configuration to be used during startup. If /autoconfig is specified, the default configuration name is used and no startup messages are displayed.

If the /autoconfig=config parameter is included in the startup command, then the configuration specified by config is used and no startup messages are displayed.

If the /noautoconfig parameter is specified or the parameter is omitted, the system prompts for the name of the configuration to be used. If the system startup routine is invoked, it displays information similar to the following:

```
MSM-UNIX, Version 4.4
Copyright (C) 1984-1998, Micronetics Design Corporation

License...: Serial# 1400000, 128 Users
Supplier..: Micronetics Design Corporation
End-User..: John Q. Customer

Enter Startup Configuration <Name>
```

In response to this prompt, enter the *Name* of the configuration to be used to initialize the MSM system. Enter a null response (the RETURN key) to use the default configuration name displayed in brackets (for example: DFLT). A configuration name identifies a collection of system definitions which describe the operating environment. For additional information on configurations, refer to “Generating the System” in this manual.

Once a configuration name is entered, MSM begins its initialization process, which includes building the terminal control blocks, initializing the system memory, building the resident UCI table, and initializing the device drivers.

/key

This parameter causes MSM to prompt you to enter the Activation Code, Supplier, and End-User name provided on the paper key. If an optional software feature such as MSM-NET or MSM-LAT is purchased after the original license is purchased, a new paper key is provided which activates the optional feature. Use the /key parameter to input the new paper key. The following terminal session illustrates the activation of MSM-LAT in a license that already has MSM-NET.

```
# ./msmstart /key

MSM-UNIX, Version 4.4.0
Copyright (C) 1984-1998, Micronetics Design Corporation

License...: Serial# 1400000, 128 Users
Supplier..: Micronetics Design Corporation
End-User..: John Q. Customer
Options...: NET(8)

Activation Code <GY1J;19UK;QWR;N;IB;EF;NYNS;BGF;H;C>:
                AP3F;W8DK;QPR;M;IB;EF;NYNS;BGF;H;C

Supplier <Micronetics Design Corporation>: RETURN

End-User <John Q. Customer>: RETURN

Please verify:
     Activation Code..: AP3F;W8DK;QPR;M;IB;EF;NYNS;BGF;H;C
     Supplier.........: Micronetics Design Corporation
     End-User.........: John Q. Customer

OK to apply <Y>: RETURN    done

Enter startup configuration <DFLT>
```

Startup Failures

When the MSMSTART shell script is invoked, it waits for a maximum of 60 seconds for the MSM system startup to complete. If the startup does not complete during this time, the shell script terminates and displays the following message:

MSM startup has failed

Please check the ./msmlog file for more details

When this occurs, the MSM initialization process has most likely detected an error condition that prevents it from completing the startup. In this case, the MSM system will write one or more error messages to the msmlog file indicating the nature of the failure. To determine the cause of the problem, review the msmlog file.

For example, if there is an insufficient number of semaphores configured in the system, then one of the following messages will appear in the msmlog file:

seminit: get semaphores failed; err=nnn

or

no space left on device

This problem typically occurs because the number of devices configured in MSM is too large for the number of semaphores available in the UNIX system. To resolve this problem, increase the appropriate semaphore values and reconfigure the UNIX kernel. Alternatively, MSM can be started by including the -t parameter on the msmstart command to decrease the maximum number of devices.

Once MSM is up and running, the number of devices can be reduced through the SYSGEN utility program. Refer to “Generating the System” in this manual for information on the SYSGEN utility. Refer to “Configuring UNIX” in this manual for information on the relationship between the number of devices and the number of semaphores required. This chapter also describes the recommended semaphore value settings.

Another common cause of system startup problems is MSM’s inability to access the System Volume Group (the first Volume Group to be mounted by MSM). MSM uses the System Volume Group to determine the configuration parameters and the remaining Volume Groups to be mounted. When this condition occurs, the following messages will be contained in the msmlog file:

mumsm: boot volume group is not mounted

mumsm: aborting

This problem can occur in any of the following situations:

- If the user who invoked the MSMSTART shell script does not have root privileges, then startup fails. To obtain root privileges, the user must either log on to the system under the root user ID or must issue the UNIX super user (su) from a UNIX shell prompt.
- This failure can occur if the MSM database has been moved, damaged, or deleted. At startup, MSM obtains the location of the database from the CONFIG.MSM file. If the database has been moved, the CONFIG.MSM will contain incorrect information, To correct this condition, either move the database back to its original location or update the CONFIG.MSM file with the new information.

- An error can occur if an attempt is made to start MSM while it is already active. In this case, the following message appears in the msmlog file:

  WARNING: MSM appears to be active

  If MSM is already running, MSMSTART will detect this condition and prevent a second copy from being started. The same condition may occur if the MSM system crashed. In this case, it may appear that MSM is running although it is not running.

To resolve this problem, first verify that MSM is actually running. Issue the UNIX ps -e command in UNIX and determine whether the muctrl, mumsm, and mudasd processes are running. If they are running, then MSM is already active. If you need to start a second copy of MSM, refer to "Starting Multiple Copies of MSM" in this chapter for instructions.

If the muctrl, mumsm, and mudasd processes are not active, then MSM probably terminated abnormally without proper cleanup. Before proceeding, report the system failure to the system manager so that diagnostic action can be taken. After this is done, it will be necessary to remove the appropriate msmbase file before MSM can be restarted.

The msmbase file normally is stored in the directory in which the MSM-UNIX distribution files are stored (mumsm, mudasd, muctrl, and so on). When the msm command is invoked, it searches the current directory for the msmbase file. An MSM option allows you to specify a different location for the msmbase file. Use caution and ensure that the proper msmbase file is removed before restarting MSM. Refer to "Starting Multiple Copies of MSM" in this chapter for information.

To delete the msmbase file, use the UNIX rm command. The msmbase file should never be removed unless MSM has crashed and only after the system manager or technical support representative has obtained any diagnostic information available to use in determining the cause of the crash.

MSM Sign-on Options

After the MSM system is started, users can log on using the msm command. The msm command supports an optional parameter that allows sign-on information to be supplied. The format of the MSM command is as follows:

```
msm SignonParm
```

SignonParm

This parameter is used to specify sign-on information to the system and is valid whenever a user invokes the msm command to log on. The SignonParm must be in either of the following formats:

UCI{,*VolName*}:*RoutName*{:*Psize*}

^RoutName{[*UCI,Volname*]}{:*Psize*}

Where:

UCI	Name of the UCI that contains the routine to be executed.
VolName	Name of the Volume Group that contains the routine to be executed.
RoutName	Name of the routine that is to be executed or the appropriate Programmer Access Code.
Psize	Size of the partition that is to be created to execute the routine.

Using this parameter, you can be automatically signed on to the system and connected to a specified UCI and application. This approach can be used from a UNIX shell script or directly from the UNIX command prompt.

The MSM command generally is invoked from the directory in which it resides. It must be able to access the *msmbase* file. The msmbase file is created by the muctrl process in the directory containing the MSM executable files. If the MSM command cannot access msmbase, it assumes that MSM is not running and aborts the logon attempt.

You can invoke MSM from a different directory only if the MSMBASE shell variable is defined. The value that is assigned to the MSMBASE variable is the *pathname* (UNIX directory) where the msmbase file resides. To define the MSMBASE shell variable, issue the following commands from the UNIX shell prompt.

```
MSMBASE=/pathname
export MSMBASE
```

The commands shown above assume that the UNIX system is running the Bourne Shell. The format of the commands may be different if other shells (for example: csh, ksh, and so on) are being used. Refer to the documentation supplied with the UNIX operating system for additional information on creating variables in other UNIX shell environments.

Starting Multiple Copies of MSM

You can run more than one copy of the MSM system concurrently on a machine. This feature is especially useful for testing an application with a new release of MSM before upgrading the production database. When multiple copies of MSM are active, they function as totally separate and independent systems, though they can communicate with each other via MSM-NET.

Normally, a file named msmbase is created in the directory which the msm executable files are stored (mumsm, mudasd, muctrl, and so on). This file is used by the muctrl task to process login requests, logout requests, and system shutdown. If an attempt is made to start a second copy of the MSM system, the msmstart shell script will detect that MSM is already active. It makes this determination after testing for the presence of the msmbase file in the current directory. If the file exists, it is assumed that the MSM system is active.

To avoid this conflict, the executable files for the second copy of MSM must reside in a different directory. It is essential for users to invoke the msm command from the proper directory so that the login will be directed to the proper system. Alternatively, the location of the msmbase file can be specified by defining the MSMBASE UNIX shell variable as described in the previous section.

Beware of mounting the same volume group from multiple copies of MSM. If this is done inadvertently, database corruption is likely to occur.

If MSM-NET is being used, it is not possible for the two systems to share an Ethernet controller using an Ethernet link. If an attempt is made to share the controller, network errors (such as missing messages) can occur. In this situation, you must shut down and restart both MSM systems. It is possible to share the controller using UDP/IP links if each MSM system uses a different IP Port Number.

USEROPS Utility

A new utility feature allows logins and logouts to be monitored. This feature requires you to create a utility called USEROPS in the MGR UCI of the boot volume group. The USEROPS utility must have two entry points, NEWPART and ENDPART. Internally, MSM issues a DO NEWPART^USEROPS at the beginning of the login process, and a DO ENDPART^USEROPS during execution of a HALT command.

Use this feature with caution. Since the USEROPS utility is called internally before the %LOGON utility, no users will able to log on if an error occurs that is not trapped. Simply creating a USEROPS routine that is empty and does not contain line label NEWPART will prevent all logins. When this code is invoked, the principal device ($PRINCIPAL) is not open. Therefore, no WRITE or READ statements should appear in the code.

This routine runs in UCI #1 in the startup volume group and is invoked by every job, including background processes started by MSM and the first job, which is responsible for invoking the STU utility to initialize the system. For this reason, avoid accessing anything other than the MGR UCI, unless the $ZUCI function is used to verify that the application UCI is available. You also should avoid updating journaled globals unless the code verifies that journaling is enabled.

This facility may be modified in future MSM versions in order to avoid its potential pitfalls. For now, keep the USEROPS code relatively simple, and always use error trapping to prevent login problems in the event of an error in the code. It also is wise to keep a session logged in at the programmer prompt at all times when debugging this code.

System Shutdown

Before you can shut down a system in a multi-user environment, all other users must be logged off and no application jobs can be running. To initiate the shutdown process, log on to the Manager's UCI in programmer mode and enter the following command:

```
DO ^SSD
```

At the prompt, verify that you want to begin a system shutdown. After the system shutdown process begins, new user logons are prohibited. When the user who invoked SSD is logged out, the shutdown continues in the background. The shutdown is only complete when all of the msm background processes have terminated. The shutdown is complete when the message "*** MSM has terminated ***" appears at the end of the msmlog file.

At this point, all of the modified disk buffers have been rewritten to disk. If MSM is shut down without following this procedure or if UNIX is shut down without shutting down MSM, information may be lost and the integrity of the database may be jeopardized.

System Backup

The purpose of a backup is to copy the information stored on the system disk to some other type of storage media. This backup information can then be used should a system malfunction render the information stored on the system disk unusable.

It is important to plan the type of backup to be used and the frequency of backups. If a backup copy of the data must be restored, all information entered into the system after the backup was created has to be re-entered. MSM system supports two types of backup: offline and online. Several forms of offline backup are described in detail in the following sections. Online backup is described in "Resilient Systems" in this manual.

Partial Backup of the Database

Utility programs supplied with the system allow backup of selected programs and globals to an auxiliary device (for example, a diskette or tape). The utilities used to perform these functions include: Routine Save (%RS), Routine Restore (%RR), Global Save (%GS), Fast Global Save (%FGS), Global Restore (%GR), and Fast Global Restore (%FGR). For information on these utilities, refer to "Routine Utilities" and "Global Utilities" in the *MSM Utility Manual*.

Since the %RS and %GS utilities save the logical structure of the data, but not the physical structure, they can be used to unload data from one system and restore it to a different system. This allows data to be moved between the various MSM system types (MSM-PC/PLUS, MSM-UNIX, and MSM-Server for Windows), as well as other M implementations.

Performing a Backup Using UNIX Utilities

The MSM database is a standard UNIX file or raw area within the UNIX system. As such, it can be backed up using the standard UNIX utilities supplied with the system. Several standard UNIX utilities such as cpio, dd, and tar can be used, as can non-standard utilities supplied by the manufacturer.

When restoring the MSM database using one of these utilities, the user must be sure that the UNIX file size limit (ulimit) has been set high enough to allow the entire database to be restored. On some UNIX systems, the default ulimit might not be large enough to accommodate the entire MSM database. Refer to the appropriate UNIX manuals for additional information on the ulimit parameter.

Resilient Systems

Overview

This chapter describes the Before-Image Journaling, After-Image Journaling, Cross-System Journaling through DDP, and Online Backup features of MSM. Use these features to create resilient system configurations.

As users become increasingly dependent on computerized applications, the impact of system failures significantly increases. At the very least, each failure is an annoyance that users do not patiently tolerate. In the worst case, a failure can severely damage a company's ability to operate. To solve this problem, system managers look for ways to construct systems that minimize these disruptions in service.

The goal is for these systems to be continuously available to the users (non-stop operation 24 hours a day) or for the system to be quickly recoverable in the event of a temporary or permanent failure. Systems designed in this way are referred to as *resilient systems*. Within MSM, the following features are provided to assist the system manager in implementing resilient systems.

Before-Image Journaling	Ensures structural integrity of a database in the event of a temporary system failure.
After-Image Journaling	Used to update a database after restoration from a temporary or permanent system failure.
Cross-System Journaling	Maintains a duplicate database on a secondary system.
Online backup	Allows database backup operations to occur while users are logged onto the system and database updates are occurring.

Each of these features operates independently of the other features. They can be used in configurations that contain a single computer or in configurations with multiple computers connected through a network. The following sections describe each of these features in detail.

Before-Image Journaling

Before-Image Journaling is used to ensure database integrity in the event of a soft failure. This optional feature, often referred to as *bullet-proofing*, records images of database blocks before modifications are made to them (for example: by a SET, KILL, or block split). These images, which are recorded in a separate host file (the Before-Image Journal file, or BIJ), are used by the system to rollback the database to a point in time where structural integrity is known to exist. As a result, Before-Image journaling restores the database to its state at some point in the past, such as a few seconds or a few minutes prior to the failure.

Once Before-Image Journaling is enabled, its operation is automatic. Whenever a bullet-proofed database is mounted (manually, or automatically at system startup), MSM checks the BIJ file to determine whether recovery is necessary (system stopped without the database being properly dismounted). If so, then the database is automatically rolled back to an earlier consistent state using the images in the BIJ file. If After-Image Journaling also is being used, the system automatically rolls the database forward using information from the After-Image Journal. The database will then be current up to the last entry in the After-Image Journal.

Getting Started

Bullet-proofing can be selected on a volume group by volume group basis. All, none, or some volume groups in a system can be bullet-proofed. When creating a new volume group with the DBMAINT utility, the system manager can specify whether or not the volume group is to be bullet-proofed. For additional information on the DBMAINT utility, refer to the *MSM Utility Manual*. For existing volume groups, the BIJ utility can be used to add bullet-proofing. Refer to “BIJ Utility” in this chapter for more information.

When a volume group is bullet-proofed, certain information is maintained by the system to ensure proper synchronization between the database and its associated BIJ file. Once this synchronization is established, the files remain in sync, even when the database is dismounted and remounted. Synchronization is established the first time that a bullet-proofed database is mounted or the first time that it is mounted after re-enabling bullet-proofing for the volume group. Refer to “Enable Before-Image Journal” in this chapter for information on using the JRNL utility to re-enable bullet-proofing for a database.

Within the database itself, synchronization information is maintained in the map blocks. If synchronization between the volume group and the BIJ file is lost, the system manager can re-establish it. Refer to “Before-Image Journal File Maintenance Options” for information on using the JRNL utility to re-establish synchronization for a database.

Although bullet-proofing is automatic once enabled, the system manager must provide certain information to initially establish Before-Image Journaling. The name and size of the BIJ file and the length of a generation within the BIJ file must be specified. The following paragraphs describe these items.

BIJ file name

The BIJ file is a standard Unix file that is used to store the before-images of blocks in a volume group. Each bullet-proofed volume group must have its own unique BIJ file (a BIJ file cannot be shared by volume groups). When bullet-proofing a volume group, the system manager is prompted for the name of the BIJ file. The name that is specified is arbitrary and may include an optional path name (for example: /mumps/msm/SYS.BIJ). The default name is /dir/VOL.BIJ, where VOL is the name of the volume group and dir is the directory of volume 0 of the volume group.

BIJ file size

It is essential to allocate sufficient disk space to the BIJ file. If the BIJ file is too small and becomes full, performance will be adversely affected. The size of the BIJ is directly proportional to the frequency of updates to the database and inversely proportional to the localization of the updates. If just a few blocks are updated repeatedly, less BIJ space is used because the before-image of each block is captured only at the first modification of the block in any generation.

Because there is no direct correlation between the static size of a volume group and size of the BIJ file, it is not possible to provide an exact formula for computing the amount of BIJ space required for a given database. In addition, the localization of updates to a database varies considerably from application to application. For systems with less than 32 users and a database of less than 512 MB, a file size of 10 MB is probably sufficient. For very busy systems with multi-gigabyte databases, a BIJ file of 40 MB or larger may be needed for optimum performance. The minimum BIJ file size that can be specified is 5 MB.

Generation length

Whenever a database update operation (SET, KILL, or ZSAVE) takes place, the before-images of all database blocks affected by the update are written to the BIJ file. This is done before any actual updates to the database take place. The smallest level of granularity in bullet-proofing is a single database update. For greater efficiency, multiple updates to the database are grouped together to form a *generation*. Any update to a bullet-proofed database is completely contained within a generation (an update does not span generations). At the point where a generation in the BIJ file ends and a new generation begins, the structural integrity of the database is guaranteed to be consistent.

At any point in time, before-images from the volume group are being written to one and only one generation within the BIJ file. This generation is referred to as the current generation, which will exist for a user-specified length of time or until a system-determined number of blocks are written to the generation. When either condition occurs, the system changes the generation.

When all the updates associated with a given generation other than the current one have been successfully written to the database, the generation can be closed. If any older generations are still open, then the system delays the close until the previous generations are closed. When the generation is finally closed, the space in the BIJ associated with the generation is freed and can be reused for future generations. In this way, the BIJ file can be considered to be a circular file. This means that all new updates in the current generation are added to the head of the file while all blocks belonging to closed generations are removed from the tail of the file.

To ensure good system performance and reliability, it is important to determine how often to change generations. For example, if After-Image Journaling is not being

used, then generations should be rather short since recovery will only rollback the database to the last complete generation. Therefore, the longer the generation, the more information that will be lost in the event of a system failure.

Generations should not be too short, however, since generation changes incur a certain amount of overhead. Conversely, allowing generations to remain open too long might require a larger BIJ file and a potentially lengthy recovery time in the event of failure.

The system permits generation lengths to be specified in seconds. The size of a generation can be from 16 seconds to 3600 seconds, with the default generation size being 600 seconds. If After-Image Journaling is used, the default should be satisfactory for most applications. If After-Image Journaling is not being used, then a shorter generation length should be specified to minimize loss of data due to rollback. The recommended default in this case is 30 seconds.

It is important to point out that the specified generation length is only an approximation to the actual generation length at runtime. The system will change the current generation more quickly if it can do so without impacting performance or if other factors in the system require that the generation be changed. The system will also delay a generation change if it detects that a forced generation change at a particular time will adversely affect performance.

BIJ Utility

Use the BIJ utility to control all functions related to Before-Image Journaling. Some functions apply only to mounted volume groups, while others may also be directed at volume group files that are initialized but not mounted.

To use the BIJ utility, log on to the Manager's UCI in programmer mode. Other users may be logged on to the system while this utility is in use. When the utility is invoked, the following menu is displayed.

```
>D ^BIJ

        MSM - Before-Image Journal Utility

Select Before-Image Journal Option

    1 - Enable Before-Image Journal
    2 - Disable Before-Image Journal
    3 - Show Before-Image Journal Information
    4 - Number of BIJ/OLB Buffers
    5 - Before Image Journal File Maintenance Options

Select Option:
```

Enable Before-Image Journal

Use this option to enable Before-Image Journaling for a volume group or to re-enable Before-Image Journaling for a volume group after the journal has been disabled. It may be used for a mounted volume group or for a volume group that is not mounted. If the volume group is mounted with Before-Image Journaling disabled, the enable function does not take effect until the volume group is dismounted and remounted.

When Before-Image Journaling is enabled for the first time, the system allocates one block from volume 0 of the volume group to be used as a status block for the journal. The volume group label is updated to point to the status block, a Before-Image Journal file is allocated and formatted, and its name is stored in the Before-Image Journal status block.

When Before-Image Journaling is re-enabled for a volume group, the Before-Image Journal file is formatted. The first time the volume group is mounted after an enable or re-enable operation, the mount procedure scans all the map blocks in the volume group to ensure that no obsolete information remains from prior bullet-proofing.

```
Select option: 1 - Enable Before-Image Journal

Enter Volume Group: ?

    Enter the name or number of a mounted volume group, or
    Enter the host file name of an unmounted volume.
    Enter ^L for a list of mounted volume groups.
    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

Enter Volume Group: LIB

Enter Before-Image Journal file name </msm/prod/LIB.BIJ>: ?

    Enter the name of the host file to contain the BIJ records.
    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

Enter Before-Image Journal file name </msm/prod/LIB.BIJ>: RETURN


File size in megabytes <10>: ?

    Enter the size of the journal file, in megabytes.
    Valid values are from 1 through 256.
    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

File size in megabytes <10>: RETURN

Ready to format? <Y>:

100% done
Format complete
Journal will be ENABLED the next time the volume group is mounted.
```

Disable Before-Image Journal

This option disables the Before-Image Journal functions for a volume group (makes it non-bullet-proofed). Bullet-proofing may be disabled for a mounted or unmounted volume group. If the volume group is mounted with Before-Image Journaling enabled, the disable function does not take effect until the volume group is dismounted.

This option is primarily intended to be used when bullet-proofing is to be permanently disabled for a volume group. However, if there is damage to the Before-Image Journal file or for some other reason the Before-Image Journal file cannot be accessed, then this option can be used to temporarily disable bullet-proofing for a volume group. When the problem is corrected, bullet-proofing can be re-enabled.

If the Before-Image Journal is temporarily disabled, a warning message will be issued each time the volume group is mounted. The user must verify that the volume group may be mounted without journaling. When bullet-proofing is permanently disabled, the system manager must delete the BIJ file associated with the volume group (the system does not automatically delete it).

```
Select Option: 2 - Disable Before-Image Journal

Enter Volume Group: ?

    Enter the name or number of a mounted volume group, or
    Enter the host file name of an unmounted volume.
    Enter ^L for a list of mounted volume groups.
    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

Enter Volume Group: LIB
Temporary or Permanent disable? ?

    If temporary disable is selected, a flag will be set in the Before-
    Image Journal status block so that the next time the volume group
    is mounted, the Before-Image Journal will be disabled and a warning
    to this effect will be issued at all subsequent remounts of the
    volume group.  The journal file will continue to be associated with
    the volume group.
    Permanent disable breaks the connection between the volume group
    and the journal file.  The file name will be removed from the
    Before-Image Journal status block.

    Enter 'T' to temporarily disable the journal.
    Enter 'P' to permanently disable the journal.
    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

Temporary or Permanent disable? T
Journal will be DISABLED the next time the volume group is mounted
```

Show Before-Image Journal Information

This option displays information about the status of Before-Image Journaling for a mounted volume group or for a volume group file that is not mounted.

```
Select Option: 3 - Show Before-Image Journal Information

Enter Volume Group: ?

    Enter the name or number of a mounted volume group, or
    Enter the host file name of an unmounted volume.
    Enter ^L for a list of mounted volume groups.
    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

Enter Volume Group: LIB

Before-Image Journal file name: /mumps/prod/LIB.BIJ
Before-Image Journal file size: 10 MB
Generation length: 600 seconds
Journal is ENABLED
```

Number of BIJ/OLB Buffers

This option is used to select the number of Before-Image Journal (BIJ) buffers and the number of Online Backup (OLB) buffers to be allocated during system startup. Refer to “BIJ/OLB Buffers” for additional information, including guidelines for determining the number of buffers to allocate.

```
Select Option: 4 - Number of BIJ/OLB Buffers

Enter Number of BIJ/OLB Buffers <4>: ?

    Enter the number of buffers to be allocated during system startup.
    There should be one buffer for each active volume group which is
    bullet-proofed, plus one for each concurrent online backup that
    will be active, plus 50% more to allow for I/O overlap.  If Bullet-
    proofing is used, a minimum of four buffers should be allocated,
    even if the calculation above produces a number smaller than four.
    The system supports a maximum of 64 buffers.

    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

Enter Number of BIJ/OLB Buffers <4>: 5
```

Before-Image Journal File Maintenance Options

This option is used to perform various system management functions that may be needed to maintain the Before-Image Journal function.

```
Select Option: 5 - Before-Image Journal File Maintenance Options

Select Before-Image Journal Maintenance Function:

    1 - Change Before-Image Journal File Name
    2 - Reset Generation Numbers in Maps
    3 - Set Maximum Generation Time

Select Option:
```

Change Before-Image Journal file name

This option allows the system manager to change the name of the Before-Image Journal file in the Before-Image Journal status block of the volume group. Use this option if it is necessary to move or rename the file while retaining the integrity of the file contents. It may be used for a mounted volume group or for a volume group file that is not mounted. This option does not actually rename or move the BIJ file itself. The system manager must actually rename or move the file (which should be done with the volume group dismounted).

```
Select option: 1 - Change Before-Image Journal File Name

Enter Volume Group: ?

    Enter the name or number of a mounted volume group, or
    Enter the host file name of an unmounted volume.

    Enter ^L for a list of mounted volume groups.
    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

Enter Volume Group: LIB

Current Before-Image Journal file name: /mumps/test/test.bij

Enter new Before-Image Journal file name: ?

    Enter the new Before-Image Journal file name.

    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

Enter new file name: /usr/bijcopy
```

Reset generation numbers in maps

This option scans all the maps of a mounted volume group and resets (marks as available) any blocks that are prevented from being allocated while waiting for the Before-Image Journal to change to the next generation. This ensures that all free blocks are made available for allocation. Under normal circumstances, the system automatically maintains synchronization of the map blocks. Use this option if the files must be re-synchronized.

```
Select Option: 2 - Reset Generation Numbers in Maps

Select volume group: ?

    Enter the name or number of a mounted volume group, or
    Enter ^L for a list of mounted volume groups.
    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

Select volume group: LIB

Ready to reset LIB? <Y>: Y

2 of 130 maps updated
```

Set maximum generation time

Before-Image Journaling must periodically pause while all database updates that are in progress are allowed to finish. This process is known as a *generation change*. Between generation changes, data accumulates in the Before-Image Journal file, which is used to recover the database in the event of a system crash.

The system closes out a generation when the system becomes quiescent, the journal file is nearly full, or the maximum generation time is reached. A generation change usually takes less than one second to complete, so users generally notice no impact on response time.

This option may be selected for a mounted volume group or for a volume group that is not mounted. If the volume group is mounted, the maximum generation time entered takes effect immediately.

```
Select Option: 3 - Set Maximum Generation Time

Select volume group: ?

    Enter the name or number of a mounted volume group, or
    Enter the host file name of an unmounted volume.
    Enter ^L for a list of mounted volume groups.
    Enter ^ to return to the previous question.
    Enter ^Q to quit the utility.

Select volume group:  LIB

Maximum generation time <600>: ?

    Enter the maximum time (in seconds) that a journal generation will
    stay open.  The value must be between 16 and 3600, inclusive.

Maximum generation time <600>: 30
```

Mount Database With Unreadable BIJ

When a bullet-proofed database is mounted, the associated BIJ file is read to determine if the database needs to be recovered. Under normal circumstances, this operation and any associated recovery is transparent to the user. If the BIJ file is unreadable, then user intervention is required. For example, if the BIJ file was inadvertently removed or overwritten (for example: by the Unix rm or cp commands), then the system cannot determine if database recovery is required. If the BIJ is unreadable, the mounting procedure for volume group 0 will differ from the mounting procedure for all other volume groups.

Mount volume group 0 with unreadable BIJ

If volume group 0 is bullet-proofed but its BIJ file is unreadable, normal system startup will not proceed, and MSM issues a message indicating the nature of the error. If this occurs, restart the system in baseline mode and use the BIJ utility to disable bullet-proofing for volume group 0. After disabling bullet-proofing, shut down the system and restart it normally. Use the VALIDATE utility to verify that the database is intact. To subsequently re-enable bullet-proofing, use the Enable Bullet-proofing option of the BIJ utility. The next time volume group 0 is mounted (the system is restarted), bullet-proofing will be in effect.

Mount other volume groups with unreadable BIJ

If you attempt to mount any volume group (other than volume group 0) that is bullet-proofed but has an unreadable BIJ file, then mounting is aborted and a error message is issued. Mounting can subsequently be performed manually using the BIJ and DBMAINT utilities. Use the Disable Bullet-proofing option in the BIJ utility to temporarily disable bullet-proofing. Since the database is unmounted, enter the host file name in response to the "Volume Group" prompt.

After bullet-proofing is enabled, mount the database using the DBMAINT utility, and then use the VALIDATE utility to ensure that the database has integrity. Use the Enable Bullet-proofing option in the BIJ utility to re-enable bullet-proofing, and then use the DBMAINT utility to mount the database. After it is mounted, bullet-proofing will be in effect. Refer to the *MSM Utility Manual* for additional information on the DBMAINT and VALIDATE utilities.

After-Image Journaling

After-Image Journaling is the method used to record information about global SET and KILL operations. The information captured during each SET or KILL operation is stored in a host operating system file. In the event of a system failure, the After-Image journal provides information necessary to automatically re-process all or part of the global SET and KILL operations that occurred since the last system backup.

The system manager may activate After-Image Journaling for all globals on the system, all globals in a specific UCI, or a specific global in a specific UCI. Refer to the *MSM User's Guide* for additional information on globals and journaling.

Invoking the JRNL Utility

To use the JRNL utility, the system manager must be logged onto the Manager's UCI in programmer mode. Other users may be logged on to the system while this utility is being used. When the utility is invoked, the following menu is displayed:

```
>D ^JRNL

          MSM - Journal Utility
           9:39 AM  9-APR-98

Journal file directory is not defined.
Journaling not active

Select Journal Option:

    1 - Activate (Start) Journaling
    2 - Deactivate (Stop) Journaling
    3 - Show (Display) Journal Spaces
    4 - Configure Journaling
    5 - Mark Journal Space as Reusable
    6 - Switch to New Journal Space
    7 - Restore (Dejournal) From Journal Space
    8 - Print a Journal
    9 - Cross-System Dejournaling Options

Select Option: ?

    Select option by specifying the option number or supplying enough
    characters to uniquely identify the option,
    To get help information specific to an option, enter a '?'
    followed by the option number.

    Enter <RETURN> or '^' to exit with no action.
    Enter '^L' to get a list of options

Select Option:
```

Configure Journaling

Use this option to create journal areas and assign a directory on the hard disk where the journal files are to be stored.

```
Select Option: 4 - Configure Journaling

Select Journal Configuration Option:

    1 - Assign Journal Directory
    2 - Create New Journal Space
    3 - Delete Journal Space

Select Option:
```

Assign Journal Directory

Before journal areas can be created, you must assign a Unix directory in which all journal files will be stored. When journal areas are created, they are placed in the specified directory. Journal control files, as well as log files that contain a history of journaling activity, are stored in the journal directory.

```
Select Option: 1 - Assign Journal Directory

Enter the directory name for journal files: ?

    Enter directory name where all journal files will reside.
    Enter <Return> or "^" or "^Q" to exit without changes.

Enter the directory name for journal files: /JOURNAL

Creating directory
```

Create New Journal Space

Before journaling can be used, you must create one or more journaling spaces. The journal spaces are host operating system (UNIX) files which are placed in the journal directory.

Define as many journal spaces as needed. Generally, you should create at least two journal files so that if the first journal file becomes full, the system can switch to the second journal file. The total amount of journal space (the sum of the sizes of all journal files) must be sufficient to handle the maximum amount of journaled data.

```
Select Option: 2 - Create New Journal Space

The file /JOURNAL/jrnlctrl is not found

File /JOURNAL/jrnlctrl now created

The new journaling file name will be: J_EMPTY.001

File size in Megabytes: ?

    Enter the size in Megabytes of the journal file to create.

File size in Megabytes: 10

Formatting.

New journal space 'J_EMPTY.001' now available

#  Status  Size   SETs   KILLs  TSTARTs  TCOMMITs  TABORTs  %Full
-  ------  ----   ----   -----  -------  --------  -------  -----
1  EMPTY     10      0       0        0         0        0    0.0

   File name=J_EMPTY.001

Journaling not active
```

Delete Journal Space

Use this option to delete a journal space that is no longer required. If a journal space was never used (it is marked Empty), there are no implications to deleting it. If the journal space was used, the remaining journal spaces are renumbered. This can cause confusion if a restore is necessary before the next full backup. If journal spaces are deleted, it is essential to keep a detailed record of the event.

```
Select Option: 3 - Delete Journal Space

Journal space name to be deleted: ?

    Select journal space by name or number.
    Enter '^L' for list of currently defined journal spaces.

Journal space name to be deleted: ^L

#  Status  Size   SETs  KILLs  TSTARTs  TCOMMITs  TABORTs  %Full
-  ------  ----   ----  -----  -------  --------  -------  -----
1  EMPTY     10      0      0        0         0        0    0.0
   File name=J_EMPTY.001
2  EMPTY     10      0      0        0         0        0    0.0
   File name=J_EMPTY.002
3  EMPTY     10      0      0        0         0        0    0.0
   File name=J_EMPTY.003

Journaling not active

Journal space name to be deleted: 3

Are you sure <N>: Y

Journal space 'J_EMPTY.003' now deleted

 #  Status  Size  SETs  KILLs  TSTARTs  TCOMMITs  TABORTs  %Full
--  ------  ----  ----  -----  -------  --------  -------  -----
 1  EMPTY     10     0      0        0         0        0    0.0
    File name=J_EMPTY.001
 2  EMPTY     10     0      0        0         0        0    0.0
    File name=J_EMPTY.002

Journaling not active
Note: Journal file numbers were changed.
```

Activate (Start) Journaling

For journaling to occur, a journal space must be active and have sufficient space to contain the entry to be journaled. Use the Activate (Start) Journaling option to start journaling. When this option is selected, the system automatically assigns the first available journal space.

```
Select Option: 1 - Activate (Start) Journaling

Journaling is now enabled

Journaling is enabled, $J=6, name=J930409.001
#SETs(0), #KILLs(0), #TSTARTs(0), #TCOMMITs(0), #TABORTs(0)
%Full=0.0
```

Deactivate (Stop) Journaling

At any time, you can stop system journaling by deactivating the current journal space. All in-progress journal activity completes, and no additional journaling occurs until the journaling process is restarted. Unlike older releases of MSM, if a SET or KILL is executed on a journaled global while journaling is deactivated, the operation waits until journaling is reactivated. Do not deactivate journaling when there are users on the system who may be updating journaled application data.

```
Select Option: Deactivate (Stop) Journaling

OK to stop journaling? <N> Y

Journaling is now disabled
```

Show (Display) Journal Spaces

Use this option to display the status of all journal spaces defined to the system. The display includes: the file name of the journal space, its status (Empty, Full, Active, Current, or Reusable), its size in megabytes, and the current number of transactions (by type) which have been recorded in the journal space.

Empty journal spaces are those that have never been used.

Full journal spaces have been completely filled with journal transactions. Before they can be reused, they must be marked *Reusable*.

The *Active* journal space is the one to which the system currently is writing. The Active journal space is marked *Current* if journaling has been deactivated. When a journal becomes full, the journal process automatically switches to the next empty or reusable journal space and marks it Active.

```
Select Option: 3 - Show (Display) Journal Spaces

Journal directory is /JOURNAL

#  Status  Size   SETs   KILLs  TSTARTs  TCOMMITs  TABORTs  %Full
-  ------  ----   ----   -----  -------  --------  -------  -----
1* ACTIVE    10   2150      20        0         0        0    5.5
   File name=J930409.001
2  EMPTY     10      0       0        0         0        0    0.0
   File name=J_EMPTY.002

Journaling is active, name='J930409.001'
```

Mark Journal Space as Reusable

A full journal space must be reset before it can be used for journaling. This prevents the journal data from being accidentally overwritten. After journal spaces are backed up or a full backup is taken, mark them as Reusable; otherwise, all journal spaces will eventually become full. If this occurs, all journaled activity will stop until you either create a new journal space or mark one as Reusable.

When this option is selected, the utility steps through each full journal space and asks if it is to be marked as Reusable. To mark a journal space as Reusable, enter **Y**(ES).

```
Select Option: 5 - Mark Journal Space as Reusable

Journal space J930409.001 will be marked as REUSABLE.
Are you sure <N>: Y

Journal space J930409.002 will be marked as REUSABLE.
Are you sure <N>: Y

All available journal spaces are now marked as REUSABLE
```

Switch to New Journal Space

When a journal space becomes full, the journaling system process (the JRNDAEMN utility) automatically marks the active journal space Full and switches to the next available empty or reusable journal space. It is not necessary for you to manually switch journal spaces. Any jobs that are updating journaled globals are suspended for one second while the journal spaces are switched. Since this is an infrequent event, impact on system performance is negligible.

You may want to close out (mark as Full) a journal space that is not yet full so that it can be backed up at a convenient time. If so, use this option to manually switch the active journal space.

This option does not actually switch journal spaces, but merely closes out (marks as Full) the current or active journal area. The system journal process (the JRNDAEMN utility) detects that the active journal has been closed, and it switches to the next available empty or reusable journal space.

```
Select Option: 6 - Switch to New Journal Space

OK to switch active journal area <N>? Y

Journaling is enabled, $J=17, name=J930409.002
#SETs(8750), #KILLs(280), #TSTARTs(0), #TCOMMITs(0), #TABORTs(0)
%Full=22.8

Changing ACTIVE status to FULL for file: J930409.002

Status change complete.
```

Restore (Dejournal) from Journal Space

Generally, the first step is to restore the last system backup file (refer to the MSMDR and OLB utilities) before restoring the journal. You then can use the Restore option of this utility to restore all or part of one or more journal spaces.

Enter the name of the journal space to be used for the restore and specify whether all or only selected journal entries are to be restored, and whether entries are to be verified as they are restored.

```
Select Option: 7 - Restore (Dejournal) From Journal Space

Dejournaling will start from J930409.001 <Y>: ?

    Enter <RETURN> or (Y)es to start dejournaling with the first
    journal space listed in "jrnlhist".
    Enter (N)o to select a different journal space.
    Enter '^Q' to exit

Dejournaling will start from J930409.001 <Y>: RETURN

Dejournal All or Selected globals? <A> ?

    Enter 'S' to restore only selected globals from the journal.
    You will be asked to select each UCI and all globals within
    it to be restored.
    Enter <RETURN> or 'A' to restore all data in the journal.
    Enter '^' to backup to the previous question.
    Enter '^Q' to exit this utility.

Dejournal All or Selected globals? <A> RETURN

All of the data will be restored
```

```
Do you wish to verify each restored SET or KILL? <N> ?

    Enter (Y)es to verify each SET and KILL operation before executing
    it.
    Each selected global entry in the journal will be displayed on your
    screen and you will be asked whether to restore each one.

    Enter <RETURN> or (N)o to restore data without asking.
    Enter '^' to backup to the previous question.
    Enter '^Q' to exit this utility.

Do you wish to verify each restored SET or KILL? <N> RETURN

Do you wish to print the data being restored? <N> ?

    Enter (Y)es to print the global data as it is restored.
    You will be asked to select the device for printing.

    Enter <RETURN> or (N)o to restore without printing.
    Enter '^' to backup to the previous question.
    Enter '^Q' to exit this utility.

Do you wish to print the data being restored? <N> RETURN

De-journal from area J930409.001
Create date/time.................: 55584,50997 (3/8/1997  2:09 PM)
Last used date/time..............: 55616,36984 (4/9/1997 10:16 AM)
File size.......................:  10,485,760 (10.0 MB)

Ready to restore? <Y> RETURN

Restoring, please wait...
Dejournaling completed for journal area J930409.001

Continue dejournaling from next area J930409.002? <Y> RETURN

De-journal from area J930409.002
Create date/time.................: 55584,51018 (3/8/1997  2:10 PM)
Last used date/time..............: 55616,37032 (4/9/1997 10:17 AM)
File size.......................:  10,485,760 (10.0 MB)

Ready to restore? <Y> RETURN

Restoring, please wait...
Dejournaling completed for journal area J930409.002

Dejournaling finished.
```

Print a Journal

Use this option to print the contents of a journal space. You can print the entire journal or selected globals.

```
Select Option: 10 - Print a Journal

Select journal area: ?

    Select journal space by name or number.
    Enter '^L' for list of currently defined journal spaces.

Select journal area: ^L

#  Status  Size   SETs    KILLs  TSTARTs  TCOMMITs  TABORTs  %Full
-  ------  ----   ------  -----  -------  --------  -------  -----
1  FULL      10   3,7210    220        0         0        0   98.4
   File name=J930409.001
2  FULL      10     8750     16        0         0        0   24.2
   File name=J930409.002
3* ACTIVE     5        0      0        0         0        0    0.0
   File name=J930409.003
```

```
Journaling is active, name='J930409.003'

Select journal area: 2

Summary or Detail <D>: ?

Enter S for a summary of journal data.
     Summary data includes a list of all globals referenced,
     the number of times each global is referenced, and the
     number of references for each global as a percentage of the total.

Enter D for a detailed printout of each journal record,
     Including BIJ (Before-Image Journal) and OLB (On-Line Backup)
     Markers, and transaction processing records.

Summary or Detail <D>: RETURN

Print All or Selected globals? <A>: S

Select globals to be restored by UCI

Enter volume group <0>: 1
Select UCI: TST

Select globals ^TEST selected
Select globals ^RETURN

Select UCI: RETURN

Start at block <0>: ?

Enter a block number at which journal printing will begin.
Press <RETURN> to start at the first journal block (block 0).

Start at block <0>: RETURN

End at block <LAST>: ?

Enter the last journal block number to be printed.
Press <RETURN> to continue printing to the end of the journal data.

End at block <LAST>: RETURN

Enter output device <0>: 0

Printing from: J930409.002

Journal file name: J930409.002
Create date/time.................: 55584,51039  (8-MAR-97  2:10 PM)
Last used date/time..............: 55621,43879  (14-APR-97 12:11 PM)
File size........................:  10,485,760  (10.0 MB)
In-use size......................:      16,384  (0.0 MB)
SETs.............................:           1
KILLs............................:           1
TSTARTs..........................:           0
TCOMMITs.........................:           0
TROLLBKs.........................:           0
Re-use generation #..............:           1

Journaling activated on 14-APR-97 at 12:11 PM
K ^|"TST,SEV"|TEST  by job #2  from BET
S ^|"TST,SEV"|TEST("DATA")="1"  by job #2  from BET
End of journal

Print completed for journal area J930409.002
```

Cross-System Journaling

Cross-System Journaling is an MSM feature that provides for maintenance of a “hot standby” system. The database of the standby system is constantly updated with transactions from the primary system so that if the system fails, the backup system can be brought live with minimal down-time.

Cross-System Journaling uses the After-Image Journaling feature of MSM to maintain a record of all updates on the live system. It continuously reads from the journal files, even as they are updated; copies the updates (SET and KILL operations) to the backup system; and resets (marks as Reusable) each journal space when all of its data has been copied to the backup system.

Access the Cross-System Journaling option from the JRNL utility. When you select the Cross-System Journaling option, the system displays a sub-menu of options, as illustrated by the following terminal session.

```
>D ^JRNL

          MSM - Journal Utility
           9:39 AM  9-APR-98

Journal file directory is not defined.
Journaling not active

Select Journal Option:

    1 - Activate (Start) Journaling
    2 - Deactivate (Stop) Journaling
    3 - Show (Display) Journal Spaces
    4 - Configure Journaling
    5 - Mark Journal Space as Reusable
    6 - Switch to New Journal Space
    7 - Restore (Dejournal) From Journal Space
    8 - Print a Journal
    9 - Cross-System Journaling Options

Select Option: 9 - Cross-System Journaling Options

Select Cross-System Journaling Option:

    1 - Cross-System Journaling Parameters
    2 - Cross-System Journaling Volume Group Assignment Table
    3 - Activate Cross-System Journaling
    4 - Deactivate (Stop) Cross-System Journaling
    5 - Print Cross-System Journaling Log
    6 - Reset Cross-System Journaling Log

Select Option:
```

Cross-System Journaling Parameters

Use this option to set up the basic Cross-System Journaling parameters. Cross-System Journaling uses the DDP protocol, a feature of MSM-NET, to copy journaled data from a primary server to a backup server. You can operate Cross-System Journaling either manually (requiring no setup on the backup server) or as part of the larger Server Management facility Both facilities can be active concurrently.

```
Select Option: 1 - Cross-System Journaling Parameters

Current Cross-System Journaling Configuration Parameters
--------------------------------------------------------
Cross-System communications.........: DDP
Server Management...................: No
Manual Cross-System Journaling......: No

Enable Server Management? <N>: ?

    Answer yes to enable Server Management.  This feature is configured
    and managed using the CSMENU Utility.


Enable Server Management? <N>: RETURN

Enable Manual Cross-System Journaling? <N>: ?

    Answer yes to enable Manual Cross-System Journaling.  This feature
    is configured and managed through the options on this menu.

Enable Manual Cross-System Journaling? <N>: YES
```

Cross-System Journaling Volume Group Table

For each Volume Group in the primary system, a backup Volume Group must be assigned. All SETs and KILLs in each primary Volume Group will be duplicated in its corresponding backup Volume Group. It is assumed that for each UCI in a primary Volume Group that contains journaled globals, there is a UCI with the same name in the associated backup Volume Group.

When Manual Cross-System Journaling is enabled, you must use this option to create a table of Volume Groups that will be replicated (continually copied to a backup server).

```
Select Option: 1 - Cross-System Journaling Volume Group Assignment
Table

Enter Primary Volume Group Name: ?

Enter the name of a Volume Group (3 uppercase letters) that is to be
Cross-System Journaled.

In order for a Volume Group to be Cross-System Journaled to a backup
system, the name of its backup Volume Group must be known.  Enter the
name of a Primary Volume Group (i.e., a live Volume Group in which
journaled globals are being updated by application routines).  You will
be asked to specify its Backup Volume Group (the location to which the
globals will be copied).  Each Volume Group that is to be Cross-System
Journaled must be entered in this table.

Note that it is not necessary to configure Cross-System Journaling on
the backup server.

Enter ^L to list currently defined entries.
Enter –VGN to delete the specified entry.
Enter Primary Volume Group Name: PRA
```

```
Enter Backup Volume Group Name: ?

Specify the backup volume group to which updates to globals in volume
group PRD will be replicated by the DEJRNDDP utility.

Enter Backup Volume Group Name: BAA

Enter Primary Volume Group Name: PRB

Enter Backup Volume Group Name: BAB

Enter Primary Volume Group Name: ^L

Cross-System Journaling Volume Group Assignment Table
              Primary VGN   Backup VGN
-------------------------------------------------
                  PRA          BAA
                  PRB          BAB
```

Activate Cross-System Journaling

Use this option to start the Cross-System Journaling task, also known as the DEJRNDDP utility. Although Cross-System Journaling can be activated even if journaling is not active, both processes generally run concurrently. The Cross-System Journaling task reads the transactions out of the journal files in chronological order and applies them to the backup system.

```
Select Option: 3 - Activate Cross-System Journaling

Select journal area <J930409.001>: RETURN

OK to start Cross-System Journaling <N>? Y

Cross-System Journaling has been started
```

Deactivate (Stop) Cross-System Journaling

Use this option to shut down the Cross-System Journaling task, whether or not global journaling is active.

```
Select Option: 4 - Deactivate (Stop) Cross-System Journaling

OK to stop Cross-System Journaling <N>? Y

Cross-System Journaling has been stopped
```

Print Cross-System Journaling Log

The Cross-System Journaling background task maintains a record of all significant events, including startup and shutdown of Cross-System Journaling; the first access of a new journal space; the detection of an inaccessible backup system (which may be down for backup or another reason); and any error.

When the Cross-System Journaling task detects a problem, it issues a message to the MSM console device indicating the nature of the problem. To obtain additional information on the problem, print the Cross-System Journaling log file.

```
Select Option: 5 - Print Cross-System Journaling Log

Enter output device <7>: RETURN

14-APR-97   7:10 PM
Cross-System Journaling started as job #3
--------------------------------
14-APR-97   7:10 PM
Cross-System Journaling from journal J930409.001 at block 0
--------------------------------
14-APR-97   7:15 PM
Cross-System Journaling from journal J930409.002 at block 0
--------------------------------
14-APR-97   7:19 PM
Cross-System Journaling from journal J930409.003 at block 0
--------------------------------

------End of log file------
```

Reset Cross-System Journaling Log

The Cross-System Journaling log file grows indefinitely unless reset using this option. It is recommended that the log be reset after each system backup.

```
Select Option: 6 - Reset Cross-System Journaling Log

OK to reset Cross-System Journaling Log <N>: ?

    Enter YES to discard the current contents of the Cross-System
    Journaling log file, DEJRNDDP.LOG.
    Enter NO to continue appending messages to the end of the
    current log.
    You might want to print a copy of the log before resetting it.

OK to reset Cross-System Journaling Log <N>: Y

Log reset
```

Online Backup

Traditionally, a database is not accessible while it is being backed up. With online backup, however, an MSM database is fully accessible while the backup is taking place. Global references, SETs, KILLs, and ZSAVEs can occur with minimal impact on system performance. Online backup allows you to archive the database to a separate medium (such as a tape) while updates are made concurrently to the database. The copy of the database can be carried off-site for safekeeping and used to restore the database in an emergency.

Reasons for Backup

There are many reasons to back up a database. There is always the possibility of physical damage to the disk (referred to as a *hard disk failure*). Examples of hard disk failures include head crashes and physical damage to the device. A *soft disk failure* occurs when the hardware is still operating properly but the database is corrupt (left in an inconsistent state). Soft disk failures can occur for reasons such as system crashes or power outages.

Use the online backup tools provided with MSM to protect the database from both hard disk and soft disk failures. When used in combination with journaling, these tools provide a means of reconstructing an MSM database to its state at the point of failure. When used in conjunction with Before-Image Journaling, After-Image Journaling, and Cross-System Journaling, resilient systems can be constructed.

Online Backup Concepts

MSM provides two types of online backup, full and incremental. A *full backup* copies the entire database to an external medium. An *incremental backup* copies only the database blocks that were modified since the last backup operation. Two types of incremental backups are supported by MSM: cumulative incremental backups and serial incremental backups. A *cumulative incremental backup* copies all of the database blocks that have been modified since the last full backup operation. A *serial incremental backup* copies all of the database blocks that have been modified since the last backup of any type (since the last full or incremental backup).

The advantage of an incremental backup over a full backup is that it generally takes less time and occupies less space on the output media. In a very large database, often only a small fraction of the blocks in the database were modified since a previous backup. In this case, an incremental backup is faster because there is less data to copy. The disadvantage of an incremental backup compared to a full backup is that when a database must be restored, the full backup and one or more incremental backups must be restored.

Full backup

To achieve maximum performance, perform full online backups using the most optimum utility available in UNIX. You can specify the name of any standard or non-standard UNIX archiving program to be invoked for copying the database to external media. For assistance with specifying an archiving program for a full backup, refer to “Defining Backup Options” in this chapter.

Since database updates are allowed to take place while the full database backup is in progress, MSM must ensure that the backup image is consistent. Assume that while the backup is running, a SET causes a database block (for example, block A) to be split into two blocks (for example, block A and block B) which are then written to disk. If the database backup has already copied block A to tape but has not copied block B yet, the tape will have an inconsistent copy of the database.

To solve this problem, MSM copies all database blocks that are modified while the full backup is running into a special file, referred to as the *spill file*. As the last step of the full backup operation, the spill file is copied to the same external medium that was used to archive the original database. Together, the database image file and the spill file provide a consistent image of the database as it existed at the end of the full database backup. If a database restore is necessary, the database file is restored and then is updated using the spill file.

Incremental backup

Incremental backups are performed by a special Micronetics-supplied program called mincrolb. The mincrolb program is invoked by a host file command issued by the OLB utility.

For serial backups, mincrolb copies all blocks in the database that were modified since the last backup. For cumulative backups, mincrolb copies all blocks in the database that were modified since the last full backup. In both cases, the copied blocks are placed into a special disk file, called the *incremental file*. As with a full backup, MSM copies all database blocks that were modified while mincrolb is running into a spill file. After mincrolb completes, both the incremental file and the spill file are archived.

Naming conventions for incremental and spill files

By convention, incremental files are named INCRnnn.VOL, where VOL is the volume group name and nnn is the number of backups since the last full backup. Similarly, spill files are named SPILnnn.VOL, where VOL is the volume group name and nnn is the number of backups since the last full backup. For a full backup (nnn=000), only the file SPIL000.VOL is produced. For incremental backups (nnn is greater than 000), both an INCRnnn.VOL file and a SPILnnn.VOL file are produced.

Defining Backup Options

Before you can perform an online backup, you must provide the system with the following information: the command for performing a full backup, the commands for archiving disk files created by the backups, and the UNIX directory to be used for files produced by the backup.

In addition, the system requires at least one BIJ/OLB buffer to be allocated. Use the OLB utility to allocate BIJ/OLB buffers. Refer to "BIJ/OLB Buffers" in this chapter for information on buffers and for guidelines on determining the number of buffers to allocate.

BIJ/OLB Buffers

With bullet-proofing and online backup, certain database blocks must be copied to auxiliary files (the BIJ file for bullet-proofing and the spill and incremental files for online backup). For efficiency, individual blocks are not written directly to these files. Instead, they are grouped together (blocked) into a BIJ/OLB buffer before they are written to disk. The pool of BIJ/OLB buffers is a system resource shared by all bullet-proofed volume groups and online backups. The size of these buffers is set at 32 KB each. Each buffer may contain up to 31 database blocks, with the remaining space in the block reserved for use by MSM.

To configure the number of BIJ/OLB buffers allocated by the system, use either the BIJ utility or the OLB utility. Use the following rule to estimate the number of buffers required by the system.

Each bullet-proofed volume group requires one buffer, and each active online backup job requires one buffer. The minimum buffer allocation should be the maximum number of bullet-proofed databases that can be mounted at one time plus the maximum number of concurrent online backup operations that will be performed at one time. This number is the recommended minimum value.

For increased performance, use a larger number because a volume group can own more than one BIJ/OLB buffer at one time. For example, when a bullet-proofed volume group fills its current buffer, the buffer is posted for output to the BIJ file. Another job doing transactions against the same database will receive a fresh buffer into which it can write additional before-images.

When this occurs, the database will own two BIJ/OLB buffers. When the first buffer is written to the BIJ file, it is returned to the free pool. The number of buffers in use at any time is dynamic. As a rough estimate, the minimum number of buffers computed above should be increased by 50 percent (the number of buffers times 1.5) and if bullet-proofing is used, a minimum of four buffers should be allocated. The system supports a maximum of 64 buffers.

It is essential to allocate a sufficient number of BIJ/OLB buffers for the system. If not enough buffers are specified and the active system runs out of buffers, performance can be severely degraded. The allocation of BIJ/OLB buffers does not take effect immediately, but rather, the next time MSM is shut down and restarted.

The OLBLOG File

The OLBLOG file contains a history of the online backups. It is created by the first online backup and grows indefinitely. Use any text editor to remove old information from the file. In the following example, the list of all backups preceding the last full backup has been removed.

```
VOL 000 FULL SPIL000.VOL 27 55571,64944 18:02 2/23/1998

VOL 001 SERIAL INCR001.VOL SPIL001.VOL 28 28 55571,65092 18:04 2/23/1998

VOL 002 SERIAL INCR002.VOL SPIL002.VOL 29 29 55572,65234 18:07 2/24/1998

VOL 003 SERIAL INCR003.VOL SPIL003.VOL 30 30 55573,65326 18:08 2/25/1998

VOL 004 SERIAL INCR004.VOL SPIL004.VOL 31 31 55574,65512 18:11 2/26/1999

VOL 005 CUMULATIVE INCR005.VOL SPIL005.VOL 28 32 55575,65638 18:13
2/27/1998
```

Each line in OLBLOG represents the result of one backup. The first field in the line contains the name of the volume group. The second field contains the backup sequence number (000 for full backup, nnn for the nnn-th incremental backup). The third field contains the backup type (full, serial, or cumulative). For a full backup, the fourth field contains the spill file name (SPIL000.VOL) followed by the OLB generation number. For incremental backups, the fourth and fifth fields contain the incremental and spill file names (INCRnnn.VOL and SPILnnn.VOL), followed by the first and last generation numbers included in the backup. In either case, the remaining fields are the time of the backup in $H and text format.

Because the OLBLOG file is used by the system when a database is restored from an online backup, it is essential to back up this file each time an online backup is performed.

Invoking the OLB Utility

To use the OLB utility, you must be logged on to the Manager's UCI in programmer mode. Other users may be logged on to the system while this utility is in use. When the utility is invoked, the following menu is displayed.

```
>D ^OLB

        MSM - Online Backup Utility

Online Backup Functions:

    1 - Full backup of a volume group
    2 - Incremental backup of a volume group
    3 - Cancel an interrupted backup
    4 - Restore a volume group
    5 - Manual restore options
    6 - Define backup options
    7 - Number of BIJ/OLB Buffers
    8 - Offline backup options
    9 - Define backup volume group set

Select Option:
```

AUTOBACK^OLB

The OLB utility includes the application entry point AUTOBACK^OLB, which allows a backup to be performed silently. It must be invoked via the DO command, and it requires at least two parameters.

Syntax

DO AUTOBACK^OLB(*VGN,BACKTYPE,VERBOSE*)

Parameter	Description
VGN	A volume group name, or a volume group index, or the name of a volume group set (a super-grouping of volume groups).
BACKTYPE	One of the following values: F Full backup S Sequential backup C Cumulative (sequential backup)
VERBOSE	If non-zero, then messages will be displayed. Otherwise, the utility operates silently. If this parameter is omitted, VERBOSE defaults to silent mode.

Description

The utility returns a status variable, QF, when done. If QF=0, then the backup was successful. Otherwise, it is equal to an error code which is defined in the text at the end of the OLB utility (see label ERRSUBR).

Full Backup of a Volume Group or Set

Use this option to perform a full backup of a specified volume group or volume group set. Before a volume group can be backed up, it must be mounted.

```
Select Option: 1 - Full backup of a volume group

Enter volume group or set: ?

    Enter the name of the volume group on which to perform the
    function, or
    Enter the name of a volume group set, or
    Enter the number of a mounted volume group.
    Enter ^L to list all defined and mounted volume groups.
    Enter ^ to return to the previous question.

Enter volume group or set: CCC
Ready to back up CCC? <N> Y

Spill file opened
Issuing host command: tar cvf /dev/rmt/2n /msm/database.ccc
a /msm/database.ccc 2408 tape blocks

Database file(s) backup was successful
Closing spill file...
Spill file closed

Issuing host command: tar cvf /dev/rmt/2 /backups/SPIL000.CCC
a /backups/SPIL000.CCC 64 tape blocks

Spill file backup successful
Volume group backup complete
```

Incremental Backup of a Volume Group or Set

Use this option to perform an incremental backup of a specified volume group or volume group set. Before a volume group can be backed up, it must be mounted. When an incremental backup is performed, the incremental backup file (INCRnnn.VOL) and the spill file (SPILnnn.VOL) are created in the backup directory. Refer to "Define Backup Options" in this chapter for additional information on the backup directory. After the incremental backup is complete, the incremental and spill files are archived to tape.

```
Select Option: 2 - Incremental backup of a volume group or set

Enter volume group or set: ?

    Enter the name of the volume group on which to perform the
    function, or
    Enter the name of a volume group set, or
    Enter the number of a mounted volume group.
    Enter ^L to list all defined and mounted volume groups.
    Enter ^ to return to the previous question.

Enter volume group: CCC

Cumulative or Serial: ?

    Select the type of incremental backup to perform.

    A cumulative backup will copy all blocks that have been modified
    since the last full backup.
    A serial backup will copy only the blocks that have been modified
    since the last incremental (or full) backup.

    Enter ^ to return to the previous question.

Cumulative or Serial: S

Ready to backup CCC? <N> Y

Spill file opened

Issuing host command: ./mincrolb 6 6 /backups/INCR001.CCC
/msm/database.ccc

Database file(s) backup was successful,

Created file /backups/INCR001.CCC

Closing spill file...
Spill file closed

Issuing host command: tar cvf /dev/rmt/2 /backups/INCR001.CCC
/backups/SPIL001.CCC

a /backups/INCR001.CCC 942 tape blocks
a /backups/SPIL001.CCC 128 tape blocks

Spill file backup successful

Volume group backup complete
```

Cancel an Interrupted Backup

Use this option to cancel an interrupted backup operation. Backup operations may be interrupted by system failure, hardware failure, and other causes.

```
Select Option: 3 - Cancel an interrupted backup

Enter volume group: ?

    Enter the name of the volume group on which to perform the
    function, or
    Enter ^L to list all defined and mounted volume groups.
    Enter ^ to return to the previous question.

Enter volume group: CCC

Spill file closed - backup canceled
```

Restore a Volume Group

Use this option to restore a volume group that has been backed up using the OLB utility. Because the OLB utility uses the OLBLOG file to determine which spill and incremental files are needed to perform the restoration, the OLBLOG file must be accessible before this option is invoked.

```
Select Option: 4 - Restore a volume group

Enter volume group: ?

    Enter the name of the volume group on which to perform the
    function, or

    Enter ^L to list all defined and mounted volume groups.
    Enter ^ to return to the previous question.

Enter volume group: CCC

Full backup was on 5/10/1997 at 23:44
  spill file is /backups/SPIL000.CCC

Serial backup was on 5/11/1997 at 18:00
  backup file is /backups/INCR001.CCC
  spill  file is /backups/SPIL001.CCC

Serial backup was on 5/12/1997 at 17:25
  backup file is /backups/INCR002.CCC
  spill  file is /backups/SPIL002.CCC

You must manually perform the initial restore from the full backup. If
there are multiple volumes in the volume group, be sure to restore all
the database files.  You must also make sure that all the spill files
and incremental files listed above are available so this procedure can
read and restore from them.

Enter 'GO' when the initial restore is complete: GO
Mounting volume group for restore processing...

Enter host file name for volume group: /mumps/database.ccc

Mount successful

Beginning restore from /backups/SPIL000.CCC
Restore successful

Beginning restore from /backups/INCR001.CCC
Restore successful
```

```
Beginning restore from /backups/SPIL001.CCC
Restore successful

Beginning restore from /backups/INCR002.CCC
Restore successful

Beginning restore from /backups/SPIL002.CCC
Restore successful

Restore complete

This is a bullet-proofed database. The before-image journal file will
be reformatted.

Formatting /bjournal/bij.ccc
File size is 10 megabytes

100% done
Format complete

The after-image journal was active at the time of the last backup.

    Do you want to apply the after-image journal now? ?

    The after-image journal was active at the time of the last backup.
    If this volume group contains journaled data, the journal can be
    applied now to bring the data completely up-to-date. If you do not
    apply the journal now, you can use the 'Manual restore options'
    menu item to do it later. If you want to do selective restore from
    the journal, use the JRNL utility.

    Enter 'Y' to restore from the journal now.
    Enter 'N' to bypass the journal restore.

Do you want to apply the after-image journal now? Y

Mounting volume group for dejournal
Updating map blocks for newly formatted before-image journal.
2 of 4 maps updated.

Mount successful

Beginning dejournaling for online backup of volume group CCC
Starting at generation 10, file J930512.002, block 0, offset 24

Dejournaled  1875 SETs,  93 KILLs from J930512.002
Dejournaling complete for volume group CCC

Unmounting volume group...
```

Manual Restore Functions

Use this option to perform manual database restore functions. Its primary use is to restore a database that cannot be restored by the automated restore procedure described in “Restore a Volume Group.” When you select this option, the following sub-menu of options is displayed.

```
Select Option: 5 - Manual restore options

Manual Restore Functions:

    1 - Mount volume group for restore processing
    2 - Restore from spill/incremental files, using log file
    3 - Restore from a spill file
    4 - Restore from an incremental backup file
    5 - Reset maps and before-image journal
    6 - Restore from after-image journal

Select Option: ? 1
```

Mount volume group for restore processing

After the database files are restored from the backup medium, this operation mounts the restored volume group (which may not yet be valid) so the spill file may be applied and any subsequent incremental backups may be restored to it. The volume group will be accessible only by the job that performs this mount operation.

```
Select Option: 1 - Mount volume group for restore processing

Enter host file name for volume group: ?

    Enter the host file name for the volume group that is to be
    mounted.

Enter host file name for volume group: /mumps/database.ccc

Mount successful
```

Restore from spill/incremental files, using log file

This operation completes the restore of a volume group. Perform the following steps **before** you invoke this option.

1. Restore the database file(s) to disk.
2. Restore the spill file, which was created during the database backup, into the backup directory.
3. Restore all subsequent incremental backup files, and their spill files, into the backup directory.
4. Mount the database, using the Mount option of this utility, so the restore may proceed.

Complete the following steps, using the Online Backup log file as a guide for determining the order and names of the incremental and spill files to be restored.

1. Restore blocks from the spill file of the database backup.
2. Restore blocks from each incremental backup and its spill file.

```
Select Option: 2 - Restore spill/incremental files using log file

Enter volume group: CCC

Ready to restore CCC from 3 files? <N> Y

Beginning restore from /backups/SPIL000.CCC
Restore successful

Beginning restore from /backups/INCR001.CCC
Restore successful

Beginning restore from /backups/SPIL001.CCC
Restore successful

Beginning restore from /backups/INCR002.CCC
Restore successful

Beginning restore from /backups/SPIL002.CCC
Restore successful

Restore Function Complete
```

Restore from spill file

This operation reads a spill file and copies the M blocks to the database. After the operation is complete, the database will be valid.

```
Select Option: 3 - Restore from a spill file

Enter volume group: CCC

Enter spill file name: ?

    Enter the file name from which to restore. It must exist in the
    backup directory /backups

Enter spill file name: SPIL000.CCC

Ready to restore CCC from /backups/SPIL000.CCC? <N> Y

Beginning restore from /backups/SPIL000.CCC
Restore successful

Restore Function Complete
```

Restore from incremental backup file

This operation reads an incremental backup file and copies the M blocks to the database. The database is not valid until the spill file associated with this backup file also is applied.

```
Select Option: 4 - Restore from an incremental backup file

Enter volume group: CCC

Enter incremental backup file name: ?

    Enter the filename from which to restore. It must exist in the
    backup directory /backups

Enter incremental backup file name: INCR001.CCC
Ready to restore CCC from /backups/INCR001.CCC? <N> Y

Beginning restore from /backups/INCR001.CCC

Restore Function Complete
```

Reset maps and Before-Image Journal

After a bullet-proofed database is completely restored, the Before-Image Journal file is invalid and must be reset before the volume group can be mounted. All the maps in the volume group must be scanned to remove obsolete generation information that would prevent free blocks from being allocated.

This option scans all the maps of a mounted volume group and resets (marks as Available) any blocks that are prevented from being allocated while waiting for the Before-Image Journal to change to the next generation.

This option also ensures that all free blocks are made available for allocation. Generally, the system automatically synchronizes the map blocks. Use this option if the files must be resynchronized.

```
Select Option: 5 - Reset maps and before-image journal

Enter volume group: CCC

Ready to reset maps? <N>: Y

15 of 230 maps updated

Before-image journal file: /bjournal/bij.ccc
File size is 5 megabytes

Ready to reset journal file? <N>: Y
100% done
Format complete
```

Restore from After-Image Journal

This operation applies any SET and KILL operations that were journaled in the After-Image Journal file since the last backup.

```
Select Option: 6 - Restore from after-image journal

Enter volume group: CCC

Beginning dejournaling for on-line backup recovery of volume group CCC
Starting at generation 10, file J930512.002, block 0, offset 24

Dejournaled  1875 SETs,  93 KILLs from J930512.002
Dejournaling complete for volume group CCC
```

Define Backup Options

Use this option to define information to the system about the backup commands, archiving programs, and directory for creating files.

```
Select Option: 6 - Define backup options

Enter volume group: CCC

Online Backup Options:

    1 - Full backup command
    2 - Full backup archive command
    3 - Incremental backup command
    4 - Incremental backup archive command
    5 - Backup file directory
    6 - List defined options

Select Option:
```

Full backup command

Use this option to define the command string that is invoked to perform a full database backup.

```
Select Option: 1 - Full backup command

Enter full backup command: ?

    Enter the command string to be issued to perform a full backup of
    the database volume(s). While this command is copying the database,
    a spill file will be created which contains any blocks modified
    while the backup is active. The spill file will not be complete and
    valid until after this command returns, so it should not attempt to
    back up the spill file.

    The command may be a host (UNIX) command which will be issued via
    the $$TERMINAL^%HOSTCMD interface, or if the command begins with
    $$, it is assumed to be an M extrinsic function reference which
    will return a non-zero value if the full backup fails, or

    If the full backup must be done externally by the operator, enter
    NOTIFY for the command string. This will cause a message to be sent
    to the terminal instructing the operator to proceed with the full
    backup of the database and respond when it is finished.  If the
    command string contains the token %FILES, it will be replaced by a
    list of the host file names of the database volumes.  Also, the
    token %SPILL will be replaced by the name of the spill file, and
    %VGN by the volume group name.

    Enter a minus ('-') to delete a definition.

Enter full backup command: tar cvf /dev/rmt/2n %FILES
```

Full backup archive command

Use this option to define the command string that will be invoked to perform the archive function for a full database backup.

```
Select Option: 2 - Full backup archive command

No command defined

Enter full backup archive command: ?

    Enter the command string to be issued to archive the full backup.
    At this point, the spill file is complete, so it may be copied to
    tape.  Also, if the full backup was done disk-to-disk, it may also
    be archived to tape.

    The command may be a host (UNIX) command which will be issued via
    the $$TERMINAL^%HOSTCMD interface, or

    If the command begins with $$, it is assumed to be an M extrinsic
    function reference which will return a non-zero value if the full
    backup archive fails, or

    If the full backup archive must be done externally by the operator,
    enter NOTIFY for the command string. This will cause a message to
    be written on the terminal instructing the operator to proceed with
    the full backup archive and respond when it is finished. If no
    external action is desired to perform the full backup archive,
    enter NONE.

    If the command string contains the token %FILES, it will be
    replaced by a list of the host file names of the database volumes.
    Also, the token %SPILL will be replaced by the name of the spill
    file, and %VGN by the volume group name.

    Enter a minus ('-') to delete a definition.
```

```
No command defined

Enter full backup archive command: tar cvf /dev/rmt/2 %SPILL

Pause before issuing command? <N>: ?

    Enter NO if no pause is necessary between the database full backup
    archive and the spill file backup archive commands.

    Enter YES if there is to be a pause between the two commands (such
    as waiting for a tape to rewind and inserting the next one). The
    operator must respond before the spill file full backup archive
    continues.

    Enter a number if there should be a fixed pause between the
    commands.  The number represents the number of seconds to wait
    before continuing.

    Enter ^ to return to the previous question.

Pause before issuing command? <N>: RETURN
```

Incremental backup command

Use this option to define the command string that will be invoked to perform an incremental database backup.

```
Select Option: 3 - Incremental backup command

No command defined

Enter incremental backup command: ?

    Enter the command string that is to be issued to perform an
    incremental backup. If no command is provided, the mincrolb routine
    will be called with the command line: ./mincrolb lo hi iname
    dbfiles

    Where 'lo' and 'hi' are the backup generation numbers to be dumped,
    'iname' is the name of the incremental backup file to be created,
    and 'dbfiles' is a list of the file names which make up the
    database.

    If a command is provided, it will replace './mincrolb' in the line
    shown above.  It must, at some point, execute  mincrolb and pass it
    the parms from the command line.

    While this command is running, a spill file will be created which
    contains any blocks modified while the backup is active. The spill
    file will not be complete and valid until after this command
    returns, so it should not attempt to back up the spill file.

    The command may be a host (UNIX) command which is issued via the
    $$TERMINAL^%HOSTCMD interface, or

    If the command begins with $$, it is assumed to be an M extrinsic
    function reference which will return a non-zero value if the
    incremental backup fails, or

    If the command string contains the token %FILES, it will be
    replaced by a list of the host file names of the database volumes.
    Also, the token %SPILL will be replaced by the name of the spill
    file, %INCR by the name of the incremental backup file, and %VGN by
    the volume group name.

    Enter a minus ('-') to delete a definition.

No command defined

Enter incremental backup command: RETURN
```

Incremental backup archive command

Use this option to define the command string that will be invoked to perform the actual archive function for an incremental database backup.

```
Select Option: 4 - Incremental backup archive command

No command defined

Enter incremental archive command: ?

    Enter the command string to be issued to archive the incremental
    backup.  At this point, the spill file and the incremental file are
    complete, so they may be archived to tape.

    The command may be a host (UNIX) command which will be issued via
    the $$TERMINAL^%HOSTCMD interface, or

    If the command begins with $$, it is assumed to be an M extrinsic
    function reference which will return a non-zero value if the
    incremental backup archive fails, or

    If the incremental backup archive must be done externally by the
    operator, enter NOTIFY for the command string. This will cause a
    message to be written on the terminal instructing the operator to
    proceed with the incremental backup archive and respond when it is
    finished.

    If no external action is desired to perform the incremental backup
    archive, enter NONE.

    If the command string contains the token %FILES, it will be
    replaced by a list of the host file names of the database volumes.
    Also, the token %SPILL will be replaced by the name of the spill
    file, %INCR by the name of the incremental backup file, and %VGN by
    the volume group name.

    Enter a minus ('-') to delete a definition.

No command defined

Enter incremental archive command: tar cvf /dev/rmt/2 %INCR %SPILL

Pause before issuing command? <N>:
```

Backup File Directory

Use this option to specify the directory where backup files, including the incremental files, spill files, and log files, are to be created.

```
Select Option: 5 - Backup file directory

No backup directory defined

Enter name of backup directory: ?

    Enter the name of the directory that is to contain the backup log
    files and the spill files.

Enter name of backup directory: /backups
```

List Defined Options

Use this option to display a list of the defined backup options.

```
Select Option: 6 - List defined options

Options for volume group CCC

Full backup command.............  tar cvf /dev/rmt/2n %FILES
Full backup archive command.....  tar cvf /dev/rmt/2 %SPILL
Pause before archive command....  NO
Incremental archive command.....  tar cvf /dev/rmt/2 %INCR %SPILL
Pause before archive command....  NO
Backup directory: ..............  /backups
```

Number of BIJ/OLB Buffers

Use this option to select the number of Before-Image Journal (BIJ) buffers and Online Backup (OLB) buffers to be allocated during system startup.

Refer to "BIJ/OLB Buffers" in this chapter for additional information on BIJ/OLB buffers, including guidelines for determining the number of buffers to allocate.

```
Select Option: 7 - Number of BIJ/OLB Buffers

Enter Number of BIJ/OLB Buffers <4>: ?

    Enter the number of buffers to be allocated during system startup.
    There should be one buffer for each active volume group which is
    bullet-proofed, plus one for each concurrent online backup that
    will be active, plus 50% more to allow for I/O overlap. If Bullet-
    Proofing is used, a minimum of four buffers should be allocated,
    even if the calculation above produces a number smaller than four.
    The system supports a maximum of 64 buffers.

    Enter ^ to return to the previous question
    Enter ^Q to quit the utility

Enter Number of BIJ/OLB Buffers <4>: 5
```

Offline Backup Options

This option synchronizes offline backup activity with MSM's After-Image Journaling facility. Synchronization is necessary to fully dejournal database updates when restoration from an offline backup is performed.

```
Select Option: 8 - Offline backup options

Offline Backup Options

    1 - Mark volume group for offline backup
    2 - Unmark volume group for offline backup
    3 - List marked volume groups

Select Option:
```

Mark volume group for offline backup

Use this option to indicate that an offline volume group backup will take place when MSM is shut down. This option saves the current position within the After-Image Journal file so that database roll-forward can be performed if a restore operation is necessary. To avoid a lengthy roll-forward of journal entries that are already included in the backup, perform this option as close as possible to the actual backup.

```
Select Option: 1 - Mark volume group for offline backup

Enter Volume Group: CCC

Marked volume group CCC
```

Unmark volume group for offline backup

This option cancels a previously indicated intention to take an offline backup of the volume group. The saved position of the After-Image Journal is marked as Invalid so it cannot be used as a starting point for any later restore processing.

```
Select Option: 2 - Unmark volume group for offline backup

Enter Volume Group: CCC

Unmarked volume group CCC
```

List marked volume groups

Use this option to list all mounted volume groups that are marked for offline backup.

```
Select Option: 3 - List marked volume groups

Volume group CCC marked on JUN 28 1998 at 16:15 (journal marker valid)
```

Define Backup Volume Group Set

Use this option to define volume group sets that will be backed up as a group. When the backups are restored, data will be logically consistent across all volume groups.

```
Select Option: 9 - Define backup volume group set

Enter volume group set: ?

    Enter the name of the volume group set to be defined or changed.
    Enter -name to delete an existing set.
    Enter ^L to list all defined volume group sets.
    Enter ^ to return to the previous question.

    The name of a volume group set may be from 1 to 15 alphanumeric
    characters, except it may not be 3 uppercase characters (to avoid
    confusion with a volume group name).

Enter volume group set: abc

Defining new volume group set abc

Enter volume group list: ?

    Enter a list of volume group names, separated by commas.
    These volume groups must all be mounted when the volume group set
    is backed up.

Enter volume group list: FSA,LIB

Enter volume group set: abc

Current definition: LIB,FSA
Enter volume group list:
```

Offline Backup

In addition to the full online backup capabilities described in “Online Backup” in this chapter, full backups of the M database can be performed while MSM is shut down. This is referred to as an *offline backup*. In order to perform subsequent incremental backups, this offline activity must be synchronized with the online MSM backup activities.

This synchronization requires two steps:

1. Notify MSM before shutdown that an offline backup is intended. Use the Offline Backup options of the OLB utility (Mark Volume Group for Offline Backup) to perform this step. Refer to “Offline Backup Options” in this chapter for additional information.

2. Use the olbtstmp program to notify MSM that the backup was performed. Run olbtstmp after the backup function is complete and before MSM remounts the volume group.

   When this program is invoked, two parameters can be specified. The first parameter is used to specify the backup directory, and the second parameter specifies the MSM volume group name in uppercase characters. When the olbtstmp program is invoked, it creates a file named OLBTSTMP.VOL in the specified backup directory, where VOL is the name of the volume group. This file contains the date and time the backup was performed.

   For example, to perform an offline backup of volume group CCC (which consists of the single volume (/msm/database.ccc), first mark CCC for offline backup using the OLB utility. Shut down MSM and perform the backup using a script file similar to the following:

   ```
   tar cvf /dev/rmt/2 /msm/database.ccc
   olbtstmp /backups CCC
   ```

   You can use any backup program to back up the volume group.

Database Recovery

If a system failure occurs, database recovery can be performed from either the Before-Image Journal (if bullet-proofing was enabled) or the online backup files. The former method is the faster and simpler method. Although recovery from the Before-Image Journal usually is possible, it can fail if either the Before-Image Journal or the database is physically damaged (for example, by a head crash on the disk). In this case, the files must be rebuilt and restored from the online backup files.

Complete Recovery from a Soft Failure

By combining Before-Image and After-Image Journaling, it is possible to completely reconstruct a database. Before-Image Journaling reconstructs the database to a recent point in the past. Dejournaling the After-Image Journal entries from this point restores the database to the point of the last entry in the After-Image Journal.

Complete Recovery from a Hard Failure

By combining online backup and After-Image Journaling, you can completely reconstruct a database. Restoring the online backup files(s) reconstructs the database to the point of the last backup. Dejournaling the After-Image Journal entries from this point restores the database to the point of the last entry in the After-Image Journal.

The combination of online backup and After-Image Journaling provides a method of recovery from any hard or soft failure. If bullet-proofing is used, it is the recommended method for recovery from soft failure because it is faster and is fully automated.

Database Recovery Using Before-Image Journal

At mount, MSM's bullet-proofing software automatically scans the Before-Image Journal to see if database recovery is necessary. If so, the pre-images of the appropriate blocks are rolled back, returning the database to its state at the end of the last closed generation. If After-Image Journaling was enabled, all transactions recorded in the journal since the time of the last closed generation are dejournaled, bringing the database up-to-date. All of this occurs automatically, with no operator intervention.

Database Recovery Using the Online Backup Files

If the database is not bullet-proofed or if it cannot be recovered from the Before-Image Journal (due to damage to this file), then recovery using the online backup files is necessary. Refer to "Restore a Volume Group" and "Manual Restore Functions" in this chapter for a description of recovery using the online backup files.

Server Management

The features of resilient systems discussed in this chapter can be combined with MSM-NET and MSM-LAT to create a distributed system with 24-hour a day availability and reliability. This section provides step-by-step instructions on how to set up such a system. First, a few terms must be defined.

A *Database Server* is an MSM system that is used to maintain the application database(s) and to service database I/O requests from Client systems. A *Database Server* can either be in the Primary state (live) or the Backup state (available if the Primary Server fails). Primary and Backup Servers are always paired, each one able to serve in place of the other. *Database Server*s are also referred to as simply Primary Servers, or Backup Servers, depending on which state they are in at the moment.

An *Application Server* is dedicated to running user application routines. Users log on to Application Servers, which use the local database only as a temporary work area. All other application data is obtained through the network from one or more Primary Servers. Application Servers access the Database Servers via MSM-NET.

An Application Server can be either a user or non-user system. A non-user Application Server does not allow users to log on; instead, it is usually dedicated to CPU-intensive or disk-intensive application routines. Non-user application servers are sometimes called Print Servers, because they frequently are used to handle disk-intensive print and sort operations and to separate these functions from the application users in order to provide better performance for users.

Application routines can be stored either on the Application Servers, or on one of the Server volume groups. In either case, translation tables on the Application Servers specify the locations of all application globals.

The following sections describe how to configure Application Server and Servers and detail how they work together to form a resilient, high-availability system.

Setting Up a Primary Server

The first step in setting up a distributed resilient system is to configure one or more Primary Servers. Because the Database Servers are the repositories for the application database, the system must have sufficient disk space to contain all of the initial application data and allow for growth. In addition, disk space must be available to hold the After-Image Journal files.

There must be enough journal space to keep the Primary Server running without a Backup Server for as long as the Backup Server potentially could be down. To determine the amount of journal space needed, first determine the maximum number of days that it could take for the Backup Server to be repaired or replaced after a failure. Include sufficient journal space to handle all database updates during that period.

Once the total amount of journal space is determined, decide how to divide it. You must create at least two journal files. For Cross-System Journaling, create journal files that are large enough to hold approximately a full day's worth of SET and KILL records and to provide enough journal areas to handle the maximum anticipated down-time, plus one extra. For example, if the system could be down for as long as six days and a typical day of processing uses approximately 50 MB of journal, then create seven journal files, each of which is 50 MB in size.

Use the following steps to set up a Primary Server. If too much data exists to fit on a single server, of if placing all of the data on one server would create a disk or network I/O bottleneck, split the data into more than on server. If there would be a disk or network bottleneck with a single server, split the data so that the average number of network requests (as shown by RTHIST) is approximately the same on all servers. If the issue is merely one having enough physical disk space on a single server to store all of the data, then attempt to group the data by application.

In a multiple server configuration, it is useful to create a system diagram that reflects the location of all application globals. Visualizing the system configuration can make it easier to manage.

Set up a Primary Server

1. Create a boot volume group (install MSM).
2. Create one or more application volume groups and load the initial application globals. Using the %GCH utility, mark all application globals to be 'Always Journaled.' Turn on the "Prevent KILL" option at the same time to protect against accidental deletion of application globals.
3. Select one UCI on the volume group that will contain application routines and use the LAT utility to create a LAT service on this volume group. Using Option 9 of the DBMAINT utility, edit the parameters on-disk. Set the maximum remote access and default remote access to Read and Write, and set LAT service availability to remote only. Refer to "Configuration Utilities" in the *MSM Utility Manual* for additional information on the DBMAINT utility.

   Although a LAT service is defined on the server, MSM-LAT is not required for the Database Server licenses.
4. Create a new "production" configuration that is a copy of the DFLT configuration provided with the system. Make the production configuration the default (using Option 6 of the SYSGEN utility), and from this point on, make all configuration-specific changes in the production configuration. The DFLT configuration should be left as is, in order to trouble-shoot configuration problems if necessary.
5. Using the Automounts option of the SYSGEN utility, configure all application volume groups to be automatically mounted during startup.
6. Use the SYSGEN utility to set global defaults to Read/Write/Delete for all user groups. Refer to "Generating the System" in this manual for assistance with this procedure.
7. Enable Before-Image Journaling for all volume groups. The size of the Before-Image Journal file is load-dependent. For the boot volume group, use the minimum size allowed. For application volume groups, use 10 MB on a lightly loaded volume group, 30MB on a moderately busy volume group, and up to 64 MB or larger on an extremely busy volume group. For most systems, 20 MB to 40 MB BIJ files are appropriate.
8. Compute the amount of journal space needed. Using the JRNL utility, assign a journal directory that is on a file system with sufficient space available to hold the journal files. The journal directory should be on a separate physical disk from the application volume groups. Create the journal files.
9. Turn on Journal Autostart and Server Management Autostart (using the Autostarts and Automounts option of the SYSGEN utility).
10. Configure networking with automatic startup. Refer to "MSM-NET Services" in this manual for assistance.

11. Create another configuration called RESTORE to be used if a restore from backup is needed. This configuration should be a copy of the production configuration, with one difference: Server Management and DDP should not be autostarted.
12. Create a duplicate database and journal directory. Back up the database files on one tape, then back up the journal directory containing unused journal files on another tape. You can use the journal directory backup tape to quickly restore clean journal files onto any system that is being rebuilt after a failure, or onto a new system that is being added to the network. The Primary Server need not be backed up again under normal circumstances.
13. Repeat this process for each Primary Server to be configured. Assign a unique name to each server and each of its volume groups.

Setting Up a Backup Server

Backup Servers are nearly mirror images of their respective Primary Servers. The application areas are identical, but the system areas (the boot volume groups) have different names. To set up a Backup Server, copy all of the MSM database and journal files from a Primary Server to a duplicate system. Rename the volume groups so there are no duplicate volume group names in the entire network.

Up to this point, nothing designates either server as a Primary or Backup. This designation is made only after all Primary, Backup, and Application Servers are installed and can “see” each other via MSM-NET. This process is covered in “Setting up a Primary Server in this chapter.

Setting Up Application Servers

An Application Server is dedicated to users and user-related processes. Typically, one or more Application Servers are created to service logged-in users. As system use grows and more users are added, more Application Servers can be added to the configuration to handle the increased load.

For some applications, it is useful to have a specialized Application Server that does not allow user logon, but is dedicated to specific application processes. These processes usually involve intensive disk I/O and are segregated from the users to avoid competition for system resources. This type of client is sometimes referred to as a *Print Server* or *Task Server*.

Set up an Application Server

1. Create a boot volume group (install MSM). This volume group need not be very large. In addition to the MSM system utilities, it can contain any library (%) routines needed by the application. Although the primary application routines will reside on one of the servers (and its backup), the library routines can be maintained locally on each Application Server. This is a requirement if the Application Servers and Database Servers are using different MSM products (such as MSM-UNIX as a Database Server, and MSM for Windows NT as Application Servers). Otherwise, the library routines can reside on the Database Servers, and accessed via Routine Search Paths.

2. Create a new "production" configuration that is a copy of the DFLT configuration provided with the system. Use Option 6 of the SYSGEN utility to make the production configuration the default and from this point on, make all configuration-specific changes in the production configuration. The DFLT configuration should be left as is, in order to troubleshoot configuration problems if necessary.
3. Use the SYSGEN utility to set global defaults to Read/Write/Delete for all user groups. Refer to “Generating the System” in this manual for assistance with this procedure.
4. Configure networking, with automatic startup. Refer to “MSM-NET Services” in this manual for additional information.
5. Turn on Server Management Autostart using the Autostarts and Automounts option of the SYSGEN utility.
6. Create a Translation table that reflects the location of all application globals. If the application is installed on a single server, this step is unnecessary. In that case, the application routines and globals all reside in the same UCI on the server. If the database must be split onto more than one server (or divided into more than one volume group on a single server), a Translation table is required so that users logged into the application UCI can access all application globals transparently.
7. When the first Application Server is built, the database files can be backed up and used to create additional Application Servers that are identical to the first one. The system name (the name of the boot volume group), the LAT node name, and the first LAT service (created by SYSGEN when LAT is configured) must be changed, or problems will occur when attempting to connect via DDP or LAT to the new system. Non-user Application Servers can be treated the same way (but separately).

   As presented here, MSM-LAT is required on the Application Servers. It is possible to configure a client/server network without LAT, but certain functions cannot be as easily automated without it. For example, MSM-LAT provides load-balancing among the Application Servers (LAT evenly distributes the users across all systems that broadcast the same service). LAT services also can easily be used to direct users to the proper Database Server.

   If terminals are connected via TCP/IP, user distribution must be managed outside MSM. For example, serial terminals can be connected directly to the Application Servers. In this case, some kind of physical switch is needed to switch users between Application Servers when one is not available. Either solution significantly increases the down-time in the event of a Primary Server failure.

The Server Management Utility

After all servers are installed, use the CSMENU utility to define the relationships between the systems. You can run this utility from a single system to set up the entire distributed system.

Use the Status of Circuit option of the DDP utility to verify that each system has a circuit to all other systems. If any circuits are missing, you must correct the problem before continuing. When all systems are fully functional on the network, invoke the CSMENU utility on any system and then define each system as either a Database Server or an Application Server. The following sample terminal sessions for the CSMENU utility demonstrate this process.

```
>D ^CSMENU

       MSM - Server Management Utility

             This is Application Server CSA

Select Client/Server Option:

    1 - Define Application Servers
    2 - Define Database Servers
    3 - Print Server Tables
    4 - Switch Backup Server to Primary
    5 - Enable Automatic Failover
    6 – Verify XSYS global structure
    7 – Update Other Systems
    8 – Activate Server Management Daemon
    9 – Deactivate (Stop) Server Management Daemon

Select Option: 1 - Define Application Servers

Application Server name: ?

    Enter the system name of one of the Application Servers in this
    network.
    Each and every Application Server must be defined in this table.
    Enter -name to delete an entry.
    Enter ^L to list client systems currently defined.

Application Server name: CSC

Enable RVG LAT Services for this Application Server? <YES>: ?

    Enter YES to allow this system to broadcast LAT services for the
    Remote Volume Groups that it mounts.
    Enter NO to disable LAT services for Remote Volume Groups.

Application Server name: ^L

Application  LAT
Server       Advertisement

-----------  -----------------
 CSA            YES
 CSB            YES
 CSC            YES
 TSA            NO
```

As illustrated by the following example, servers are defined using Option 2 of the CSMENU utility.

```
Select Option: 2 - Define Database Servers

Database Server Name: ?

    Enter the system name of one of the Database Server systems in this
    network.
    Each and every Database Server must be defined in this table.
    Enter -name to delete an entry.
    Enter ^L for a list of Database Server systems currently defined.
    Enter ^ to return to the menu.

Database Server Name: FSA

Volume Groups for Server FSA <FDA,FEA>: ?

    Enter the names of the volume groups on FSA which are to be mounted
    by client systems.  Enter the volume group names, separated by
    commas, in the order in which they are mounted by the server
    system.
    Enter ^ to back up to the previous prompt.

    {Note that the default value for this prompt is the list of all
    currently mounted volume groups on the server, starting with volume
    group 1.}

Volume Groups for Server FSA <FDA,FEA>: RETURN

Server Status of FSA (Primary or Backup): ?

    Initially, each server begins as either a Primary Server or a
    Backup Server. Subsequently, server status should only be changed
    after a system failure.
    Enter P for Primary, or B for Backup.
    Enter ^ to backup to the previous prompt.

Server Status of FSA (Primary or Backup): Primary

Name of this system's Backup when in Primary state: ?

    Each server system can operate either as a Primary or a Backup
    Server. Though a system might begin as a Backup Server, a failure
    of its Primary Server will cause the Backup Server to begin
    operating as the Primary Server. Each server must know the name of
    its Backup Server.

    Enter the name of the server that will serve as the backup of this
    server when it is operating as a Primary.
    Enter ^L for a list of server systems currently defined.
    Enter ^ to return to the previous question.

Name of this system's Backup when in Primary state: SSA

Database Server Name: FSB

Volume Groups for Server FSB <FDB,FEB>: RETURN

Server status of FSA (Primary or Backup): Primary

Name of this system's Backup when in Primary state: SSB

Database Server Name: SSA

Volume Groups for Server SSA <SDA,SEA>: RETURN

Server status of FSA (Primary or Backup): Backup

Name of this system's Backup when in Primary state: FSA

Database Server Name: SSB

Volume Groups for Server SSB <SDB,SEB>: RETURN
```

```
Server status of FSA (Primary or Backup): Backup

Name of this system's Backup when in Primary state: FSB

Database Server Name: ^L

SERVER   VOLUME GROUPS   STATUS    BACKUP
------   -------------   ------    ------
FSA      FDA,FEA         Primary   SSA
FSB      FDB,FEB,FFB     Primary   SSB
SSA      SDA,SEA         Backup    FSA
SSB      SDB,SEB,SFB     Backup    FSB

Database Server Name: RETURN
```

Once all of the servers are defined, use Option 3 of the CSMENU utility to print the configuration.

You only need to define the Application Server and Database Servers on a single system. If DDP is started on all of the systems, you can propagate the configuration to the other servers. Select Option 7, Update Other Systems, from the Server Management menu. The server definition, which is stored in the XSYS global, will be copied to all of the other servers. If changes are made later, this option can be invoked again to update the other systems.

```
Select Option: 7 – Update Other Systems

Checking circuit table...

All table updates completed successfully.
```

To verify the consistency of the XSYS structure, use Option 6, Verify XSYS Global Structure, on the Server Management menu.

Server Management Operations

After the Server definitions are entered, bring the entire distributed system online by shutting down MSM and restarting it on each system. The Server Management Autostart initiates a process on each system called CSSTART, which verifies that the information it has about the network agrees with the other systems and then mounts the Remote Volume Groups containing the application database. During the startup process, the CSSTART utility temporarily disables user login to prevent anyone from using the system until all application volume groups are mounted and ready for work.

In this configuration, all users are supported through a single LAT service that exists on a single server volume group. This service is always enabled on both the Primary and Backup Servers. However, since the volume group is configured to offer its LAT services only when it is mounted remotely, the clients are the only systems that broadcast the service. When a failover occurs and the Backup Server is mounted, the same LAT service is broadcast by the client systems, but on the backup volume group.

Backups are taken of the application volume groups on the Backup Server(s) at regularly scheduled intervals (such as daily). Backups can be taken online using the Online Backup facility of MSM or offline using non-MSM backup software. While the Backup Server is down, the Primary Server stores updates in the journal files. When the Backup Server is restarted, the Cross-System Journaling process on the Primary Server detects that the Backup Server is available again, and begins sending over the accumulated data.

Since the boot volume groups of all clients and servers tend to be static (the data rarely changes), they can be backed up independently. In most cases, a backup can be taken immediately after the systems have been fully configured, and not backed up again unless changes are made to the MSM configuration. If any boot volume group is lost due to disk failure, you can restore it from its own backup.

Server management daemon

The Server Management feature uses Cross-System Journaling to keep the Backup Servers current with the Primary Servers. The Cross-System Journaling process for Server Management is controlled independently from the Manual Cross-System Journaling facility. The background task that performs the "dejournaling" is the same utility routine, DEJRNDDP, used by Cross-System Journaling. When this utility is controlled by the Server Management facility, it is called the Server Management Daemon.

When Server Management is autostarted, a set of utility routines verifies the current status of all of the servers in the network and then terminates. On Primary Servers only, the startup utilities invoke the Server Management Daemon, which then immediately begins processing journal records and copying the transactions to the Backup Server. When the system is up and running, the Server Management Daemon can be stopped and started at any time, for maintenance purposes, by invoking the options on the Server Management menu.

```
Select Option: 9 – Deactivate (Stop) Server Management Daemon

Ok to stop Server Management Daemon <N>? Y

Server Management Daemon has been stopped


Select Option: 8 – Activate (Start) Server Management Daemon

Ok to start Server Management Daemon <N>? Y

Server Management Daemon has been started
```

Journal file management

Perform a backup of the After-Image Journal files when the application database is backed up. Back up the Full and Active journal files and the journal control files (jrnlctrl, jrnlhist, jrnlog, DEJRNDDP.LOG). After the backup is complete, mark the Full journal files as Reusable either manually using the JRNL utility or programmatically using the SETSTAT entry point in the JRNDAEMN utility.

The following command is issued. *JNAME* is the host file name of the journal (not including the path):

```
D SETSTAT^JRNDAEMN(JNAME,"REUSABLE")
```

To determine if any journals are full, use the Host File Server to read-in the contents of the jrnlctrl file in the journal directory (which is recorded in ^SYS("JOURNAL_DIR"). Each line in the file is in the format "JNAME^STATUS^SIZE^data". If the STATUS field is marked Full, the journal is ready to be marked Reusable if it has just been backed up.

Note Journal files should never be marked Reusable on the Primary Server. This could cause the Backup Servers to lose their synchronization with the Primary Servers.

Backing up multiple servers

When the application database is spread across two or more servers, backups should be taken concurrently. Although it is not possible to precisely synchronize backups on multiple systems, start backups as close to the same time as possible.

Synchronizing backups can be significantly impacted by the sizes of the application volume groups. All servers should be configured identically in number of volume groups and their physical sizes; then, if backups are started concurrently, they should finish concurrently. This is especially important when using the MSM Online Backup facility.

Application transactions that involve updates on more than one system are vulnerable to being "out-of-sync" if a restore is necessary. It is possible for the updates to have been sent to one server but not to the other at the time the backups completed on the different servers (or the time that the Backup Server was shut down for backup). If an update is split in this manner, you will notice application inconsistencies when the databases are restored.

Server Failover

Cross-System Journaling is designed to maintain an online backup system that can take the place of a failed server. When a server failure occurs, failover procedures (server switch) can be initiated either automatically or by the user.

During the failover process, the following changes must be made on all of the systems that are still functional.

1. The Application Servers must unmount the volume groups on the server that is down.
2. The Application Servers must mount the volume groups on the Backup Server.
3. The Translation Table on the Application Servers must be updated to select the change in volume groups.
4. Cross-System Journaling must begin on the new Primary Server, so that when its partner system comes back up, the systems will be brought back in sync.

All of these actions are performed automatically in MSM using the Switch Backup Server to Primary option of the CSMENU utility. The following sections describe the user interface for the failover process.

User-initiated switch from Backup to Primary Server

The CSMENU utility includes an option to switch the Backup Server into a Primary Server mode. Failover is *always* initiated from the Backup Server. The following terminal session demonstrates this procedure.

```
Select Option: 4 - Switch Backup Server to Primary

Ok to make this system (SSA) the Primary Server? <NO>: ?

    Enter YES to make this server into the Primary Server and make the
    current Primary Server the Backup Server.
    Enter NO to leave this option without changing server states.

Ok to make this system (SSA) the Primary Server? <NO>: YES

The Primary Server, FSA, is still running.
```

```
Do you want to continue? <NO>: ?

    Enter YES to force all non-Manager users off of the Primary system
    before switching it to Backup mode.
    Enter NO to abort the Server status change.

Do you want to continue? <NO>: YES

Switching to Primary state

Notifying client systems: CSA, CSB, CSC

Done
```

Do not attempt a failover when users are on the system, because application errors may occur when the network connection is broken between the Application Servers and the Primary Servers. Although this feature functions when the Primary Server is running, it is intended for use when the Primary Server is down because of hardware or software failure.

Making a down Primary Server a Backup Server

When the Primary Server is down and is replaced by the Backup Server, the original Primary Server must be brought up as the Backup Server. Use the following steps to complete this process.

Bring up a Primary Server as a Backup Server

1. Restore the application volume groups to the down Primary Server from the current backup tape for its Backup Server. If the boot volume group was lost, restore it from its own backup (the backup of the boot volume group from the original Primary Server).
2. After the restores are complete, restart the down Primary Server.

The boot volume group for each server knows the name(s) of its application volume groups. During startup, if a server detects that the name of the volume group is the name of its backup, it assumes that a restore was performed and it renames the volume group before mounting it.

Since servers are passive (they do not initiate data requests from other systems), the network will not be harmed by the former Primary System coming up initially marked as a Primary Server. When the Primary Server starts its DDP servers, it checks the state of its backup. If it finds that the Backup Server is in the Primary state, it automatically switches itself into the Backup state.

From the time the Primary Server went down, the original Backup Server (which has become the Primary Server), accumulates data in its journal files. When it detects that its partner system is running again, it begins sending over the data, fully assuming the role of the Primary. In time, the two systems will again be in sync.

If both Primary and Backup Servers are equipped equally with tape drives or other backup devices, then it is never necessary to switch the roles of the two systems back to their original settings. The system that began as the Primary Server can continue to operate as the Backup Server indefinitely, and it can be backed up as the original Backup Server was backed up. However, if only the original Backup Server can be backed up, the two systems must be switched again after the Cross-System Journaling facility has brought the two systems in sync.

To perform the switch, ensure that users are logged off the system. Although the switch can be performed with users on the system, application errors can result when the switch occurs.

Automatic switch from Backup to Primary Server

If this feature is enabled using the CSMENU utility, the CSSTART utility runs continuously on the Backup Server, checking every 60 seconds to ensure that it can still reach the Primary Server. If the Primary Server is unreachable but the Backup can still reach the Client systems, then it initiates a switch (failover) without user intervention. Internally, the procedure is the same as for a user-initiated switch.

Automatic failover is enabled by invoking the Enable Automatic Failover option in the CSMENU utility, as shown in the following terminal session. MSM-LAT is required to perform a truly automatic failover, unless external software is used to automatically switch the users to the alternate system.

```
Select Option: 5 - Enable Automatic Failover

Automatic failover from Backup to Primary? <NO>: ?

    Enter YES to allow a Backup server to automatically switch to
    Primary status if the Backup loses communication to its Primary.
    Enter NO to require manual failover.

Automatic failover from Backup to Primary? <NO>: RETURN
```

Application Server considerations

Application Servers can be brought up and down at any time, regardless of the state of the Database Servers. If an Application Server is down when a switch occurs, it will attempt at restart to mount the Primary Server. When the Application Server checks the state of the Primary Server and discovers that it no longer is the Primary Server, it then checks the state of the Backup Server. After the Application Server recognizes that a switch occurred, it makes the necessary changes in its own configuration and mounts the new Primary Server.

If MSM-LAT is not being used, terminals must be diverted to an available system when an Application Server becomes unavailable.

Recovery Procedures

When any node in a distributed system fails, recovery procedures must be performed to return the system to use. These procedures vary in a client/server environment depending on the function of the failed node.

Primary Server failure

If a Primary (or Backup) Server fails due to a temporary loss of power or a temporary software problem, it is not necessary to invoke the failover process. Simply restart the down system, and processing can continue. Users who are logged in at the time of the event will experience errors and be logged off the system. Minor inconsistencies in the application data, requiring programmer maintenance, could result from such an event. The same circumstances apply during a failover. It is best to avoid a failover if the down system can simply be rebooted.

If the Primary Server failure is more permanent (for example: CPU, disk, controller), use the following steps to return the system to its original redundant state.

Recover from a Primary Server failure

1. Initiate the switch of Backup Server to Primary Server (failover). Refer to "User-Initiated Switch from Backup to Primary Server" in this chapter for step-by-step instructions.
2. Restore the data on the Primary Server from the last backup of the Backup Server database. Restore only the application volume groups. If necessary, restore the boot volume group from its own backup. Delete the contents of the journal directory and restore the journal files and their associated control files from the backup that was taken when the system was first configured.
3. Restart the down system. This server becomes the new Backup Server.
4. If the new Backup Server is equipped for backups, the recovery procedure is complete. Otherwise, schedule a brief down-time for the two servers to be switched back to their original configuration. After the final switch, immediately take a backup of the Backup Server.

Backup Server failure

Failure of a Backup Server is handled differently from that of a Primary Server. For Primary Servers, all serious hardware failures (failover, repair, restore) are treated the same. If the system cannot be restarted immediately, a failover is performed. For Backup Servers, there is no failover.

If possible, Backup Servers should also be restarted after a failure. Before-Image Journaling and After-Image Journaling protect the integrity of the database. Cross-System Journaling ensures that the two systems remain in sync.

This section describes the procedures to use when the Journal Disk, Database Disk, or CPU (including any component of the CPU, such as a network card) of the Backup Server fails.

Recover from a Backup Server CPU failure

1. Replace the CPU or affected component and reboot MSM. Cross-System Journaling ensures that no data will be lost and that the Primary and Backup Server pair will remain in sync.
2. Use the VALIDATE utility to check database integrity in case the failure affected the database. (Although Before-Image Journaling protects the database against corruption in the event of a power failure, it cannot prevent corruption when the hardware fails.) It is possible that even non-disk failures could cause the disk to write invalid data to the database. Do not assume there are no database errors because the disk drive and controller did not fail.
3. If the VALIDATE utility detects errors, you must make a decision. If there are few errors, you can determine confidently that all such errors have been identified and can be repaired manually, and that the system can be brought online afterward. (Note that MSM-NET should be shut down during the validation and repair process.)

   If the errors are more pervasive, respond as if the database disk drives failed. Refer to "Recover from a Backup Server database disk (or controller) failure" in this chapter for assistance.

Recover from a Backup Server journal disk failure

1. If MSM is still running after the failure, shut down at least DDP and LAT to prevent the Primary Server from sending more data. Perform either an online or offline backup of the MSM database files on the Backup Server.
2. Shut down MSM and replace the journal disk. Restore the empty journal files (and their associated control files) from the backup of the journal directory that was taken when the system was first created. (You may need to manually recreate the journal directory first.)
3. Optionally, if the journal directory backup is lost or destroyed, the journal directory and journal files can be recreated on the new journal disk.
4. Restart MSM. The Cross-System Journaling facility will keep the systems in sync.

Recover from a Backup Server database disk (or controller) failure

1. Replace the faulty disk or controller.
2. Restore the last backup of the volume group. If the application is divided into multiple volume groups, then *all* volume groups on *all* servers must be restored.
3. Start MSM using the RESTORE configuration.
4. Manually restore all FULL journal files and then restore the CURRENT journal file.
5. Shut down MSM and restart it in production configuration.

Application Server failure

When an Application Server fails, the results usually are not serious. Users who are logged on to the system are disconnected, and application transactions may be partially filed. However, the distributed system as a whole continues to run, and users can log on again immediately to another client. The other possible impact of losing an Application Server is that performance can suffer when the remaining clients take on additional users.

Recover from an Application Server failure

1. Attempt to reboot and start MSM normally. If the server comes up, no further action is necessary. If it does not, remove the server from the network for repair or replacement. The server can be brought back online at any time.
2. An Application Server can be rebuilt from the original backup tape. Since all data on these systems is either static or temporary, no application data is lost when restoring from the original tape. Take additional backups only after updates to the MSM operating system or to application library routines that reside on the Application Servers. For all systems, take backups of the host operating system (UNIX, Windows NT, and so on) whenever changes or updates are installed.
3. Once the system is repaired and the backup restored, restart the system. No other intervention is required.

Alternate Application Server Configuration

You can configure MSM differently than the configuration described in this chapter and still achieve the same result. The standard configuration maintains a single copy of the application routines in a single server UCI. The LAT service on this server, which is broadcast by the client systems, directs all users to this UCI.

An alternative approach is to create the application UCI on the Application Servers. In this case, the application LAT service is created on each of the Application Servers rather than on a server. For performance reasons, the server volume groups are still remotely mounted, but users do not log on to them. Although the Translation tabless on the Application Servers still indicate the location of all application globals, the “From UCI” in the tables is on the Application Server instead of the Database Server.

A disadvantage of this approach is that a full copy of the application routines must be maintained on each of the Application Servers, requiring additional disk capacity on these systems. In addition, when updates are made to the application, they must be propagated to all client systems.

One advantage of this alternative is that temporary work globals can be more simply maintained on the Application Servers, thus reducing the load on the network. To keep temporary globals on the Application Servers in the server-based application configuration, they would have to be entered in the Translation tables of each client. Since each client would have to translate these globals to a work UCI on its own system, the Translation tables would no longer be identical across all client systems. When rebuilding failed systems, be aware of this situation.

Additionally, having multiple copies of the application might make it easier to test new versions of the application code. If application modifications do not require changes to global structures, a new version of the application can be brought online on one Application Server, while the other servers continue to run the older code. Once the new version is approved, the other Application Servers can be brought down one-by-one to be updated, while the system as a whole continues to run with one less client.

The no-application server configuration

For smaller installations, a single Database Server can also act as an Application Server. In this case, the entire application (routines and globals) can reside in one UCI on one volume group (thus eliminating the need for global translation) or in several volume groups if necessary. Assuming LAT is used, the server volume group which contains the application LAT service must be configured to offer the service locally, instead of remotely. However, this LAT service must initially be manually disabled on the Backup Server.

When a failover is performed, the application LAT service must be enabled on the Backup Server. During the failover, MSM automatically enables any LAT services it finds on the non-boot volume groups (all but volume group 0). When the original Primary Server is restarted as a Backup Server, MSM automatically disables these services.

Volume Groups

Overview

In MSM, a *volume group* is the structure that is used to organize information on disk. This structure provides a method of physically combining separate independent files (Volumes) into a single logical database (a volume group).

Each volume group is a totally independent database (includes routines and globals). A volume group can have from 1 to 30 distinct User Class Identifiers (UCIs).

When the MSM system is started, at least one volume group must be present. The volume group mounted at system startup is referred to as the *System volume group*. The first UCI of the System volume group is referred to as the *Manager's UCI*. At any time after the system is started, additional volume groups can be mounted or dismounted. However, the original System volume group cannot be dismounted while the system is running.

A Volume Group consists of from one to eight separate Volumes (individual files within the host operating system). This group of files forms a complete logical database. The total size of a single Volume Group, and therefore a single database, is limited to 16,384 MB (16 GB). Although a single Volume can be a maximum of 16 GB in size, the total combined size of all Volumes within a Volume Group cannot exceed 16 GB. Although a single large Volume might be easier to maintain, you can maximize disk throughput by spreading the database across multiple disk drives.

Within a Volume Group, individual Volume sizes may be limited by constraints within the UNIX operating system. For example, in UNIX the ulimit parameter controls the maximum size of a file. MSM cannot override the maximum ulimit of the UNIX operating system. Most UNIX operating systems support files up to 2 GB in size, but some allow larger files.

Organization of Volume Groups

Each Volume Group is considered a separate and complete database. MSM-UNIX allows up to 32 Volume Groups to be mounted at one time. Additional Volume Groups can be accessed using the optional Distributed Data Processing (DDP) feature of MSM. Refer to "MSM-NET Services" for additional information on DDP. Each Volume Group that is mounted on the system contains the following information:

Volume Group Label	Block zero of the first volume in each Volume Group contains a *label block*. The Volume Group label block includes the Volume Group name, the number of volumes within the group, each volume's host operating system file name, and the total number of disk blocks in the Volume Group.
Volume Label	Block zero of each Volume contains a label block. The Volume label block includes the volume name and the total number of disk blocks in the volume.
UCI Table	A table containing all of the UCI definitions associated with the Volume Group. Each Volume Group can contain up to 30 separate UCI definitions.

Since a volume group is a complete database, all globals within the database are part of the same protection group for security purposes. Refer to "Globals" in the *MSM User's Guide* for additional information on MSM's security features.

Location of Volume Groups

Volume groups can be mounted on the local system, or they can be accessed using the DDP feature. Refer to "MSM-NET Services" in this manual for additional information on DDP.

When a volume group is mounted on the local system, the physical location of the database may be either the local system or another MSM system that is connected to the local system via DDP. When the database belongs to the local system, it is known as a *local volume group*. When the database belongs to another system, it is known as a *Remote Volume Group (RVG).*

The location of the database for a mounted volume group does not affect the way data is accessed or the type of operations permitted, except that some maintenance operations may only be performed on a local volume group.

Mounting a Volume Group

At system startup, MSM has access to the System volume group, which was created during system installation. Because the System volume group is always required for correct operation of MSM, it is permanently attached to the system (it cannot be dismounted). Because other volume groups represent independent databases, they can be dynamically attached or detached from a running system.

Use the DBMAINT utility program that is supplied with MSM-UNIX to mount or dismount any of the other volume groups. When a volume group is mounted, the UCIs associated with the volume group are also mounted. For additional information on the DBMAINT utility, refer to the *MSM Utility Manual.*

Once mounted, the UCIs can be accessed implicitly using the translation feature, explicitly using Cross-UCI notation, or by logging on to the volume group in direct mode. Refer to the *MSM User's Guide* for information on the translation feature or Cross-UCI notation.

Use the SYSGEN utility to specify which volume groups are to be automatically mounted during normal system startup. The STU utility performs the mounting of the volume group. For additional information on the SYSGEN utility, refer to "Generating the System" in this manual.

Volume Group Characteristics

The following characteristics, which pertain to each mounted volume group, describe the type of access allowed for the database and attributes of the data stored in the database:

- Whether the database is mounted for read/write access or is read-only.
- Whether other systems on a network may mount the database as a Remote Volume Group.
- The maximum length permitted for global data values.

Logging on to a Volume Group

When more than one volume group is mounted on a system, a user may need to log on to a specific volume group. To set up this logon procedure, specify the UCI and volume group that are accessed as part of the sign-on procedure. The following example illustrates how this is done:

```
MSM-UNIX, Version 4.4.0  Line #nn  UCI: UCI,VOL:PAC
```

In this example, **UCI** is the name of the UCI within the volume group, **VOL** is the name of the volume group, and **PAC** is the Programmer Access Code. Optionally, you could include the partition size for the job being created.

Storing Blocks in a Volume Group

All disk information in a Volume Group is stored using a logical block structure where each block contains 1024 characters (1K). For disk devices that do not use a 1-K physical block size, MSM converts its 1-K blocks as necessary to fit the device.

Structure of a Disk Block Number

All disk information in MSM is referenced by its logical block number. A *logical block number* is an unsigned 24-bit number that is the relative position of the block within the volume group. The first block in the volume group is 0, the second is 1, and so on. The largest possible block number that can be represented by 24 bits is 16,777,215, and therefore, the largest possible size of a volume group is 16 GB.

For convenience, most of the utility programs allow block numbers to be specified in the form Vol:Blk where Vol is the Volume number and Blk is the relative block number within the specified volume. Use this feature to organize UCIs or Globals within a single disk volume. This form of the block number is internally converted by the utility routines to the 24-bit logical block number described above. For additional information on these utility functions, refer to the *MSM Utility Manual*.

Remote Volume Groups

In addition to the volume groups described in the previous sections, the MSM system supports Remote Volume Groups. A Remote Volume Group is essentially the same as a local volume group except that it exists on another system connected through a network to MSM. The Remote Volume Group can be mounted, users can log on to the Remote Volume Group, programs can be loaded and saved, and globals can be accessed without the need for Cross-System global notation or translation. For additional information on Remote Volume Groups, refer to "MSM-NET Services" in this document.

Configuring UNIX

Overview

MSM-UNIX uses a facility within the UNIX operating system that is referred to as the Inter-Process Communication Subsystem (IPCS). This feature of UNIX includes three distinct optional components that you can select for inclusion in the system: shared memory, semaphores, and messages. Only shared memory and semaphores are used by MSM.

The default values that are assigned to the memory and semaphore options are usually inadequate for proper MSM-UNIX operation, especially for larger systems. Therefore, you must modify these settings and reconfigure the UNIX system. The AIX operating system is an exception to this rule, because the IPCS features are all configured dynamically without reconfiguring the UNIX kernel.

The procedure for reconfiguring the UNIX system should be clearly defined in the system documentation supplied by the manufacturer. Refer to that documentation for specific instructions on reconfiguring the UNIX kernel.

The following sections in this manual provide detailed information about the shared memory and semaphore requirements of MSM-UNIX, as well as the associated UNIX parameters that must be changed to accommodate these requirements.

Shared Memory Usage

Each MSM system requires that an area of memory be set aside for disk buffers, terminal buffers, system control blocks, and process communication areas. In MSM, this area is referred to as the *buffer pool* or the *buffer cache*. The size of this area is totally under your control. You can specify the size of the buffer pool using the SYSGEN utility. Refer to "Generating the System" in this manual for additional information on SYSGEN.

The memory for this buffer pool is obtained from UNIX using the shared memory feature of IPCS. Shared memory is allocated in increments called segments, each of which is assigned a unique identifier by UNIX. Other processes can access the same physical memory by acquiring shared memory with the same unique identifier. Information then can be passed between the UNIX processes that comprise the MSM system.

The following two factors must be considered when determining the size of the buffer pool:

- The primary purpose of the buffer pool is storage of disk buffers, which hold routines and globals. Each disk buffer within the buffer pool is 1K in length.

  A minimum of 32 disk buffers should be available for each partition (job) that can be started in MSM including user logon jobs, background jobs, DDP servers, and so on. With a minimum size buffer pool, MSM will operate correctly, although maximum performance may not be obtained.

  The optimum size of the buffer pool is dependent on the size and nature of the applications that will run under MSM. Generally, 128 buffers (128K) per partition are adequate. Using this guideline, a two-user system would require a buffer pool of 256K, and a 16-user system would require a buffer pool of 2048K (1 MB). The minimum buffer pool size is 512K.

  For some applications, it may be desirable to increase the size to a much larger value. This is especially true when the MSM database is very large (measured in gigabytes rather than megabytes). Generally, the more disk buffers that are available to the application, the better performance will be. As a result of the internal algorithms used by MSM, the time required to search the disk buffer cache is not affected by the size of the cache.

- Buffer pool size also is based on the number of devices (terminals, printers, magnetic tape, and so on) that will be configured in MSM. Generally, each device in MSM requires buffers (input, output, or both) and a control block to store information about the device. Memory for these areas is allocated from the same shared memory that is used for the buffer cache. As more devices are configured into MSM, less memory is available for disk buffers.

  As a guideline, each device configured in MSM requires approximately 1K of shared memory. For each terminal that is defined to MSM, the number of disk buffers that can be allocated to the cache will be reduced by one. For example, in a 16-user system, assuming 16 terminals, a 1024-K shared memory area would result in approximately 1008 disk buffers. In most cases, it is not necessary to adjust the size of the buffer pool to compensate for the number of devices.

Shared Memory Parameters

After determining the optimum size of the buffer pool, you may find that MSM is not able to allocate the necessary shared memory area from UNIX. This occurs because the UNIX parameters that control the allocation of shared memory are not large enough to accommodate the MSM request. To correct this, one or more UNIX configuration parameters must be changed, and the UNIX kernel must be regenerated.

Within the UNIX system, there are three parameters associated with shared memory that affect the operation of MSM. The combination of these parameters determines whether MSM can obtain the requested buffer pool.

The following list describes the shared memory parameters that affect operation of MSM. Generally, these parameters are the same for all implementations of UNIX. Since the names and uses of these parameters can be altered by the manufacturer, consult the appropriate UNIX system documentation before reconfiguring the UNIX kernel.

shmmax Specifies the maximum size (in bytes) of a single shared memory segment. This value should always be specified in mulitples of 1024 bytes. A value of 512K would be specified as 524288 bytes (512*1024).

shmmni Specifies the maximum number of shared memory segments that can be active in the system at one time.

shmseg Specifies the maximum number of shared memory segments that a single UNIX process may have allocated at one time.

At startup, MSM attempts to allocate the buffer pool as a single segment of shared memory. If this fails, the system attempts to allocate multiple segments that add up to the requested size. If all of the requested memory cannot be obtained, then MSM allocates the maximum amount that it can. For example, assume that the system manager has requested a buffer pool size of 6 MB for a system configured with 48 partitions (128K times 48 partitions) and that the UNIX system was set up with the following shared memory parameters:

shmmax 524288

shmmni 10

shmseg 6

In this case, MSM is not able to allocate shared memory as a single 6-MB segment (shmmax maximum is 512K). It will attempt to allocate as many segments as it can. In this case, it will only obtain six segments (shmseg is set to 6). The result is a buffer pool size of 3 MB instead of the requested 6 MB buffer pool size.

Even though the requested buffer pool size was not possible, the system will still complete startup and initialization. However, warning messages will appear in the msmlog file to indicate that MSM was unable to obtain the full amount of shared memory requested. The system manager should either modify the UNIX shared memory parameters so that the entire buffer pool can be allocated, or reduce the size of the buffer pool so the error does not occur.

There are two ways that shared memory parameters can be changed to allow allocation of the requested buffer pool. One method is to change the value of the shmseg parameter to 10. This allows 10 segments of 512K each to be allocated by MSM, resulting in a shared memory area of 6144K to be allocated.

To ensure optimum performance, it is better to have as few segments as possible. A better choice would be to change the shmmax parameter to allow a segment size of at least 6291456 bytes (6 MB). If the UNIX system allows it, the segment size should be increased to allow the entire buffer pool to reside within a single segment. Otherwise, set the value as high as possible, and increase the other parameters if necessary to allow enough segments to satisfy the request.

Semaphore Usage

The semaphore option of the IPCS facility provides a mechanism for separate user processes within the UNIX system to synchronize execution activities. Semaphores are somewhat analogous to the LOCK command in the M language. When a user has control of a semaphore and another user requests the same semaphore, the requesting user must wait. When the user who owns the semaphore releases it, the waiting user gains control of the semaphore and execution resumes.

The number of semaphores required by MSM is determined by the number of terminal devices that can be active within the system, and by the number of muserver processes. The maximum number of terminal devices is set by the SYSGEN utility whenever terminals are defined or edited, or the maximum number of partitions is set. The number of muserver processes is also set by SYSGEN. Assuming that the maximum number of terminal devices is N, and the number of muserver processes is M, then the number of semaphores required by MSM is 2N+M+8 (two times the number of devices plus the number of muserver processes, plus 8).

Semaphore Parameters

Four parameters associated with semaphores affect the operation of MSM. The combination of these parameters determines whether or not MSM can allocate enough semaphores for defined devices, as well as the semaphores needed by the system itself. These parameters are described below.

semmns Specifies the total number of semaphores in the system. For MSM to operate properly, this value must be at least 2N+M+8, as described above.

semmni Specifies the maximum number of semaphore identifiers (semaphore groups) that can be active in the system at one time.

semmsl Specifies the maximum number of semaphores that can be obtained per semaphore identifier.

semmnu Specifies the total number of semaphore undo structures in the system. This value should be equal to the value of the semmns parameter.

During system startup, MSM-UNIX examines the CONFIG.msm file to determine the maximum number of devices (ports), and the number of muserver processes. Based on this information, it attempts to obtain the required number of semaphores from UNIX. If enough semaphores are not allocated in UNIX or the value of the semaphore parameters prevents MSM from successfully allocating the necessary number of semaphores, then the following message appears in the msmlog file:

```
seminit: get semaphores failed
```

If this occurs, MSM automatically decreases the number of devices so that system startup can complete. Normally, the startup utility (STU) will report that an insufficient number of semaphores prevents some terminals from being configured.

A minimum of 15 semaphores must be available in UNIX. If not, then the MSM-UNIX system startup is terminated and an error message is sent to the msmlog file.

If the value of the semmnu parameter is too small, then MSM not start. When this occurs, the message "no space left on device" appears in the msmlog file. If the semmnu parameter is too small, MSM may start, but an attempt to logon will fail at some point. When this occurs, the same error message is displayed on the terminal attempting to log on to the system.

For example, assume that MSM-UNIX was configured to include nine UNIX shared devices, four MSM dedicated devices, one parallel device, and four muserver processes. This configuration requires a total of 40 semaphores (2 times 14 devices, plus 4, plus 8). Assume that the UNIX system was set up with the following semaphore values:

semmsl	25
semmni	10
semmns	60
semmnu	60

At system startup, MSM-UNIX will obtain one semaphore identifier that contains 25 semaphores and one semaphore identifier that contains 16 semaphores. This provides a total of 40 semaphores (2*14+4+8) required for the configuration.

Now assume that an additional 20 terminals are defined to the system, bringing the total number of devices to 34. MSM would then require 90 semaphores in order to support all of the configured devices. In this case, the semmns and semmnu parameters must be increased to 100.

There is no adverse effect on the system if these parameters are set larger than actually required. It is a good idea when reconfiguring UNIX to set the parameters to allow for future expansion. Refer to "Reconfiguring the UNIX Kernel" in this chapter for additional information on configuring the UNIX kernel.

Reconfiguring the UNIX Kernel

If the parameters for shared memory or semaphores need to be changed, then for most UNIX operating systems, you must modify the UNIX configuration file and regenerate the UNIX kernel. Although the configuration parameters are often found in a file named *conf.c* in a subdirectory of the root file system, the location of this file and the procedure for modifying it varies between hardware systems. For instructions on reconfiguring the UNIX kernel, consult the appropriate documentation supplied by the hardware manufacturer.

Issuing UNIX Commands

Overview

MSM provides a mechanism that allows you to issue UNIX commands from programmer mode or from within a program. When MSM is operating in the UNIX environment, you may want to issue UNIX commands without exiting MSM. For example, an MSM program may need to start a UNIX background task at a specific time each day.

The MSM system provides three distinct modes in which UNIX commands can be issued: terminal commands, foreground commands, and background commands.

Terminal Commands

When a UNIX command is issued in this mode, all terminal activity from MSM is suspended and the input/output operations from the invoked command process are directed to the terminal. When the command finishes, MSM resumes control of the terminal.

This could be used, for example, to invoke the UNIX visual editor program (vi) to edit a file. While in the editor, all input from the terminal is sent to the editor and all output from the editor is sent to the terminal. When editing is complete, control returns to MSM.

To accomplish this, the system first saves all of the terminal settings that are currently in effect for MSM. It then reinstates the UNIX terminal parameters that were in effect when the user logged into MSM. The specified command is then called. When the command terminates, the MSM terminal parameters are restored. UNIX commands that alter the terminal characteristics (for example, stty) will have no effect once control is returned to MSM, because MSM reinstates the terminal options in effect at the time the command was called.

Foreground Commands

When a UNIX command is issued in this mode, the MSM program is suspended until the command has completed. The terminal remains connected to MSM (it is not connected to the command that was invoked) while the UNIX command is being performed. Use this mode when the application must wait until the UNIX command is completed, but no I/O is required from the UNIX process.

Since the terminal is still connected to MSM, any terminal input or output operations performed by the UNIX command will be ignored. MSM accomplishes this by setting the standard input (stdin), the standard output (stdout), and the standard error (stderr) devices to use the /dev/null device. In this mode, an input request returns an immediate end-of-file indication and output to the device is discarded.

Background Commands

When a UNIX command is issued in this mode, the M job that issued the command is not suspended and the terminal remains connected to MSM. The MSM program continues to run while the UNIX command is being executed. No indication is given to MSM when the UNIX command completes.

A command issued in this mode could be useful, for example, to create a UNIX process that will transfer files created by MSM to another host system and programming language. Assuming that the actual time at which the transfer takes place is not important to the program that issued the command, then a background UNIX process would be the most appropriate.

Input and output operations in this mode are treated in exactly the same manner as they are for foreground commands. Since no input or output will occur to the terminal, the UNIX stdin, stdout, and stderr devices are redirected to the /dev/null device.

Escaping to the Shell

One method of issuing UNIX commands in MSM is from programmer mode. This is referred to as escaping to the UNIX shell. A UNIX command that is issued from programmer mode can be either a terminal command or a background command. It is not possible to issue a foreground command from programmer mode. The following shows how this is done:

```
! UNIX command
```

In programmer mode, when the first character of input in response to a command prompt is an exclamation point (!), the system passes all input that follows, up to the line terminator character, to the UNIX shell as a command. This is always assumed to be a terminal command and will be processed as described above. The following examples illustrate how UNIX commands can be entered from programmer mode:

```
>!pwd
>! ls -lt
>! vi /users/fred/myfile
>!tar c /usr/msm/* &
>! nohup >/tmp/tar.log tar cv /usr/msm/* &
```

To edit an M routine using the UNIX vi editor, use %RS to save the routine to a file named temp, and enter !vi temp at the programmer prompt (> !vi temp). Use %RR to restore the routine from the file named temp.

A command which ends in an ampersand (&) indicates to UNIX that the command should be executed as a background process. This allows you to issue Background Commands while in programmer mode. When a command is issued in this manner, it acts like a Terminal Command that terminates instantly; however, the process that is created continues to execute.

Foreground processes run as a child process of the msm process that invoked the login, and they run in the directory from which msm was invoked. The process honors the UNIX file access permissions of the UNIX user that invoked msm. Background processes are invoked from muserver, and run in the directory from which msmstart was run. They do honor the user's access permission.

Issuing Commands from a Program

All three types of UNIX commands can be issued from within an MSM routine through use of *extrinsic functions* supplied with the system. One extrinsic function exists for each mode in which a UNIX command can be issued. These functions are contained in the %HOSTCMD utility program. The following shows how each function can be invoked.

UNIX Command Types

Extrinsic Function Entry Point	Command Type
`SET X=$$TERMINAL^%HOSTCMD("UNIX Command")`	Terminal command
`SET X=$$JOBWAIT^%HOSTCMD("UNIX Command")`	Foreground command
`SET X=$$JOB^%HOSTCMD("UNIX Command")`	Background command

In each of these examples, the value that is passed to the function is the UNIX command to be executed. Refer to the *MSM Language Reference Manual* for additional information on extrinsic functions and how information can be passed to the function.

Since the UNIX commands are executed as a result of a call to an extrinsic function, a value is returned to the caller. For terminal commands and foreground commands, the value that is passed back is the return code that was received from the command that was issued. Any meaning assigned to the value is specific to the command being executed. Typically, if the return code is zero, the command completes normally (without an error).

For a background command, the value returned is the process identifier for the shell process that invokes the UNIX command being executed. Since no indication is given when a background command terminates, this is the only available information about the new process.

The other difference between these command types is the UNIX environment in which the UNIX command executes. For terminal commands, the environment is the same as the UNIX user that originally logged on to MSM. The UNIX environment is handled the same as when escaping to the UNIX shell. This shell process terminates immediately upon creating a sub-process for the UNIX command. Therefore, the value does not indicate the id of the UNIX process that ultimately executes the command.

Foreground and background commands, however, are handled differently. These commands are executed from a shell that is created by one of the muserver processes. As a result, the UNIX command operates in the same user environment as MSM-UNIX itself. MSM passes the proper user privileges to the command shell (the command will be executed as if it was invoked by the user), but its present working directory (pwd) is the one from which msmstart was invoked.

Because the working directory can be different depending upon the type of UNIX command being issued (terminal, foreground, and background), any file names that are referenced should include the full UNIX path name. This prevents confusion about which file is being accessed.

Memory Management

Overview

This chapter describes how MSM-UNIX handles the diverse memory needs of the multi-user M operating system. Several distinct subdivisions of MSM memory are described and instructions for tuning the system for optimum performance are provided.

The Buffer Pool

The *buffer pool* is used as a cache for all disk I/O operations. When MSM requires a disk block, it searches the buffer pool to determine whether the block is currently in memory. If it is, the request is immediately satisfied. If the requested block is not in the buffer pool, then it must be read from disk. When this occurs, execution of the job is delayed while the I/O takes place.

The MSM buffer manager retains blocks in the cache based upon the *most recently used* algorithm. This ensures that the routines and global blocks needed most often by the application usually reside in memory. This feature makes it unnecessary to explicitly force certain routines to be memory-resident, as is true in other systems.

The size of the buffer pool should be based on the number of user partitions and the size of each partition. As a general rule, since all disk blocks (both routine and global blocks) are maintained in the cache, the larger the buffer pool, the better the performance. When determining the size of the buffer pool, each defined partition should have a minimum of 128K.

For larger systems, the size of the database is more significant in determining the buffer pool size than the number of users. In general, for systems in which MSM is the only application process being used, the buffer pool should be set to 30 to 50 percent of the total physical memory.

The MSM Partition

Each MSM job or task has an associated area of memory called the *partition*. The partition contains local variables and routines that are being edited. Only when a routine has been explicitly loaded into the partition via the ZLOAD command does it use any memory in the partition. Since routines actually reside in the buffer pool while they are being executed, they require no partition memory while in this state.

If an attempt is made to get memory beyond the limit set by the partition size, a <PGMOV> error occurs. You can select any default partition size from 12K through 16M using the SYSGEN utility. Refer to "Generating the System" in this manual for additional information on this utility.

The Partition Stacks

Each job has two special areas of memory that are allocated for performing certain system functions. The first area is the Execution Stack or simply, the *stack*. This area of memory is devoted to keeping track of program nesting which occurs when the DO and XECUTE commands are executed. The second area is the String Accumulator Stack or simply, the s*tap*. It is used to store intermediate results during the evaluation of complex expressions such as nested function calls, concatenation, indirection, and so on. It also is used by the ZSAVE command to store the p-code generated during compilation.

By default, both stacks are 20K in size. While the default stap and stack sizes are adequate for most applications, some applications may require the stack and the stap sizes to be increased. For example, if an application becomes deeply nested through repeated use of the DO and EXECUTE commands, then the stack size must be increased to avoid <STKOV> errors. Similarly, if a large routine generates a <STKOV> error when it is being compiled by the ZSAVE command, then the stap size must be increased to prevent the error. Use the SYSGEN utility to increase the stack and stap sizes, as needed, to a maximum size of 256K each.

Configuring Memory in SYSGEN

The SYSGEN utility includes two options that affect the way MSM allocates memory. The Default Partition Size option is a configuration-specific parameter which indicates the maximum amount of memory that can be assigned to any partition. The Buffer Pool Size option is a system-specific parameter which indicates how much memory is to be set aside for the disk cache.

Default Partition Size

To determine the appropriate Default Partition Size for the system, first identify the maximum amount of partition memory required by the application. To do this, run the application on one terminal while periodically running the %SS utility on another terminal. Use the %SS utility to determine the maximum partition size used by the application by inspecting the current partition size each time the utility is run and noting the largest value.

It is helpful if a programmer who is familiar with the application is available to specifically invoke the software options that are known to require the most local variables. If a <PGMOV> error occurs, use either the SYSGEN utility or the %PARTSIZ utility to increase the partition size. Rerun the program until the errors no longer occur.

After you determine the maximum size, round it up to the next multiple of 5K and add 5K for a safety margin. In SYSGEN, use this value to set the Default Partition Size for the production configuration. Refer to "Generating The System" in this manual for additional information on the SYSGEN utility.

Buffer Pool Size

The size of the MSM buffer pool is the single most important factor in overall system performance. Refer to "The Buffer Pool" for a complete description of how the buffer pool is used. Buffer pool size can either be the default value or can be determined by experimentation.

In either case, if the Cache Read Hit Ratio displayed by the RTHIST utility is less than 95%, you generally can increase throughput by increasing the buffer pool size. Refer to "Performance Monitoring" in the *MSM Utility Manual* for additional information on the RTHIST utility.

Although the Disk Cache Efficiency displayed by the %SS utility tends to slightly underestimate the true caching efficiency of the system, it is reasonably accurate. Use the SYSGEN utility to enter the new buffer pool size. The next time the MSM system is started, the new buffer pool size will be used.

Tape Devices

Overview

This chapter describes MSM's support for industry-standard 9-track tape drives, cartridge tape drives, and Digital Audio Tape (DAT) drives. Information on tape naming conventions and features is provided.

The MSM system supports tape cartridge devices and 4mm and 8mm DAT tape devices on most UNIX systems. For certain systems, 9-track tape devices also are supported. Users can create tapes in ASCII or EBCDIC, with ANSI or IBM labels, in different recording formats, and at different recording densities. For more information, refer to "Using Peripheral Devices" in the *MSM User's Guide*.

Tape Naming Conventions

As with other devices, tape devices in UNIX are referenced by their device names. When defining a tape device (devices 47, 48, 49, and 50) in SYSGEN, you must identify the UNIX device name to be associated with the device (for example: /dev/rmt0, /dev/rct). The device name for a tape actually is a file name contained in the /dev directory of UNIX. This file name describes the device and its attributes. Although there is no universally accepted standard for tape device names, many hardware manufacturers use the same naming conventions for the devices.

For example, 9-track tape devices typically include the letters mt in the name to indicate that the device is a magnetic tape. Cartridge tape devices usually include the letters ct to indicate that the device is a cartridge tape. On some systems, the letters mt are used to indicate all types of tape devices.

In addition, every tape device generally has at least two device names, including one name that can be used to access it as a block mode device (for example: /dev/mt0) and one name that can be used to access it as a raw mode device (for example: /dev/rmt0). In most cases, the only difference between the device names is the use of the letter r as a prefix. In order for MSM to properly access a tape device, the raw mode device name must be used. If the block mode device name is used, errors will occur.

On some systems, the tape device file names (or nodes as they are sometimes called in UNIX), exist as a separate directory under the UNIX /dev directory (they appear as a subdirectory). In this case, the file name itself does not include the 'rmt' or 'rct', but instead, the name of the subdirectory of /dev is 'rmt' or 'rct'. For example, /dev/rmt/0 might indicate unit 0 of a raw mode 9-track tape device.

The following table lists the tape device names used by many of the UNIX systems supported by MSM. If the system you are using is not listed in the table, consult the UNIX documentation supplied with the hardware to determine the appropriate tape device name.

Tape Device Names

Model Name	Cartridge Tape Name	9-Track Tape Name
DG Aviion	/dev/rmt/0	/dev/rmt/0
Digital UNIX	/dev/rmt0h	Not supported
Hewlett Packard	/dev/rmt/0m	/dev/rmt/0
IBM RS/6000	/dev/rmt0	/dev/rmt0
SCO UNIX	/dev/rct0	Not supported
Siemens RM400	/dev/ios0/rstape006h	Not supported
Solaris	/dev/rmt0	Not supported

The digit 0 in the device names in the table above generally refers to the tape device unit number. When only one tape device is installed, then 0 always is used. When multiple tape devices are installed, this digit varies depending on the unit number of the specific tape device.

In UNIX, tape density (for example: 1600 BPI) and rewind options (for example: rewind on close) are controlled using an alternate tape device name. For example, on most systems, the device name includes the letter h to indicate high density, the letter m to indicate medium density, or the letter l to indicate low density. This letter usually is appended to the end of the device name. When defining a tape device in SYSGEN, this letter is not included as part of the name. MSM automatically selects the appropriate tape device name, depending on the tape density selected on the OPEN or USE command.

On many systems, the letter n appended to the UNIX device name indicates that no automatic rewind of the tape is to occur when it is closed. As with tape density, MSM internally selects the appropriate device name to control the rewind operation when the tape is closed. When defining a tape device in SYSGEN, this letter is not included as part of the name.

Tape Device Restrictions

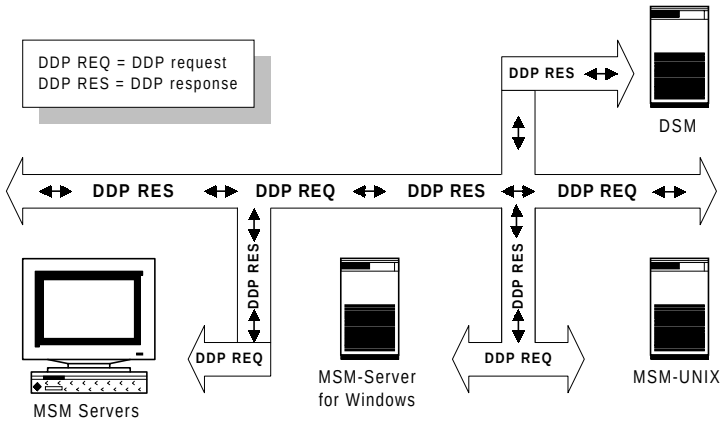
MSM-UNIX controls tape devices by using software instructions in the UNIX operating system. MSM supports only features that are supported by UNIX. Because these restrictions depend on the release level of the UNIX operating system, you should consult the README file in the MSM distribution package for information on any current restrictions.

MSM-NET Services

Overview

MSM-NET is an optional package for MSM-UNIX. MSM-NET provides services that enable MSM systems to network with other MSM systems and other M Technology systems. The MSM-NET package supports the Remote Volume Group (RVG) protocol between MSM systems, the DDP protocol between MSM, DSM, and ISM systems, and the Open M Interconnect (OMI) protocol to OMI systems. The following figure illustrates the use of MSM-NET in a mixed-machine environment with the Distributed Data Processing (DDP) protocol.

MSM Networking in a Mixed Environment



The DDP protocol allows a job executing on one M system to access global data on another M system. The protocol used for DDP requests (the format of the messages sent on the network) can be specific to a particular M implementation, or it can be the Open M Interconnect (OMI) protocol, which allows unlike systems to exchange data. MSM-NET supports an efficient protocol for communication between MSM systems. It also directly supports the Digital Standard MUMPS (DSM) and Intersystems protocols, as well as the OMI protocol.

All DDP processing is based on a request-response transaction. A job executing on the local system sends a request for a particular global operation to the remote system and waits for a response. The remote system receives the request, performs the desired operation, and returns an appropriate response. In some cases, when using the MSM protocol, requests and responses may be bundled into a single transaction to improve network efficiency. In DDP terminology, the system making the request is called the *client* or agent, and the system processing the request and returning the response is called the *server*.

The Remote Volume Group protocol provides extended capability for sharing data between MSM systems. It allows one MSM system to mount a volume group that actually exists on another MSM system. When the volume group is mounted, not only is global data available, but the volume group is available for users to log on to the UCIs it contains and to execute routines. All data block accesses are maintained in the system's local buffer cache, reducing the need for repeated network activity when subsequently accessing data in the same block.

A Remote Volume Group is handled in nearly all cases the same as a local volume group. The primary difference is that access to the database is provided by MSM-NET instead of the local MSM database manager. Refer to "Volume Groups" in this manual for additional information on using volume groups.

The physical network that connects the systems together is typically Ethernet (IEEE 802.3) or Token Ring (IEEE 802.5). Other physical networks can be used if they are bridged to Ethernet transparently through hardware. DSM systems can only communicate with MSM systems using standard Ethernet networks.

MSM-NET Distribution Package

Although all components of the optional MSM-NET package are included in the standard distribution for MSM-UNIX, MSM-NET is enabled within the paper key only if it has been purchased.

There is no installation procedure for MSM-NET per se, since all of the utility programs are loaded into the database when MSM is installed, and the network drivers for most systems are built-in to MSM. Simply use the SYSGEN utility to configure MSM-NET.

For certain systems (for example: the IBM RS/6000), additional driver modules may need to be installed with the MSM-NET package. Generally, it also is necessary to install certain UNIX software modules (for example: TCP/IP) before the MSM-NET option can be used. Refer to "Installing MSM-NET" in this chapter for additional information on these requirements.

Networking Components in MSM-NET

The basic components of an MSM network include the physical and logical links, nodes, and circuits. The following sections provide an overview of these components.

Links

In order for two or more systems in a network to pass information between each other, there must be a physical communication path between them. This path is commonly referred to as a Communications Link, or simply as a *Link*. Generally, this physical Link can be established using communications hardware. The most common types include Ethernet and Token Ring.

MSM-UNIX supports Ethernet in what is commonly called *native mode*. In other words, MSM can send raw Ethernet packets. Native mode provides the highest level of network throughput, and is required for native mode protocols such as LAT (Local Area Transport).

MSM also supports a number of layered network protocols and interfaces. Layered protocols can be used to communicate over Wide Area Networks (WAN). The following sections provide an overview of the native and layered link types supported by MSM.

Native Ethernet links

MSM-UNIX supports the IEEE 802.3 standard for Ethernet communication. In MSM, links that adhere to this standard are referred to as *Ethernet links* and are considered physical links. Each node connected to an Ethernet link includes an Ethernet controller and an Ethernet transceiver. The physical connection between nodes is accomplished by connecting the transceivers with coaxial cable. In many modern Ethernet cards, the transceiver is built into the controller, eliminating the need for a separate external device.

By design, every manufactured Ethernet controller has a unique address. The address consists of 12 hexadecimal digits and generally is shown as six groups of two-digit numbers separated by hyphens (for example: AB-CD-EF-12-34-56).

When a message is sent to a node, the Ethernet address of the controller on the remote system is used to specify the location. With the exception of broadcast and multicast messages, a controller only receives messages that contain its unique controller address. Every message received by a node includes the controller address of the node that sent the message.

Layered network protocols and interfaces

MSM-UNIX supports a variety of high-level network protocols that are independent from the type of network used (Ethernet). Such high-level protocols can be used to communicate over a Wide Area Network.

TCP/IP is possibly the most widely supported network protocol. UDP/IP is an IP-based protocol similar to TCP/IP. The TCP/IP protocol is used for Open M Interconnect, or OMI, a standard protocol for communicating between different M implementations. UDP/IP can only be used for MSM to MSM communications.

Nodes

When multiple processors are connected together in a network, each individual processor is referred to as a *Node*. Every node in the network has a database that consists of one or more volume groups. The volume groups that comprise the database are said to be “mounted” on the node. Each node may have up to 32 separate volume groups mounted at one time. Refer to “Volume Groups” in this manual for additional information.

Circuits

Just as a physical connection between machines is required, a logical connection between the MSM systems is also required. This logical connection is referred to as a *circuit*. Within a node, a Circuit table is constructed that includes one entry (circuit) for each remote node in the network that can be accessed. The circuit includes information such as the Ethernet Controller address, a list of volume groups mounted on the node, and the name of the node.

Operational Considerations for MSM-NET

Beyond the physical network, several factors influence the way in which MSM-NET operates. The factors discussed in this section affect the absolute system performance, overall network management strategy, and programming methodology.

DDP Servers

The MSM system creates one or more separate jobs for processing requests to access globals and routines across the network. Referred to as *DDP Servers*, these background jobs execute MSM utility programs that handle global SETs and fetches across the network from other systems, LOCK commands, error conditions, startup and shutdown of nodes, and Automatic Network Configuration.

When DDP usage is relatively light, a single DDP server is usually sufficient to handle the network traffic. As DDP traffic increases, jobs may be delayed because a single DDP server cannot process all service requests quickly enough. When this occurs, overall system performance may be degraded.

To overcome this problem, MSM-NET allows multiple servers to operate concurrently. The DDP workload then is distributed among the servers, allowing multiple DDP transactions to be processed in parallel. This approach increases system throughput and minimizes the time required to satisfy requests made by individual jobs.

At most, there should be one DDP server, plus one DDP server for each circuit that is actively being used. Circuits that are lightly used can be ignored. The number of DDP servers used is only one of several factors that influence overall system performance.

Automatic Network Configuration

MSM's Automatic Network Configuration feature simplifies network management. When this feature is used, every startup, shutdown, and reconfiguration of a node is automatically detected by all other nodes that are connected to the network. When a node is started, it broadcasts a message to all other nodes indicating that it is coming online to the network.

The message sent by the new node also notifies other network nodes which volume groups it has mounted. In response, all other network nodes send a message to the node that is starting up, indicating their node name, node address, and a list of the volume groups they have mounted. Similar messages are sent whenever a volume group is mounted or dismounted and whenever a node is shut down.

Automatic Network Configuration honors Network Security if it has been specified, and DDP groups if they are defined. For example, if Network Security has been specified, passwords are included in the broadcast message. Systems receiving the broadcast message check their security list to ensure that the password matches. If not, no acknowledgment is sent to the broadcast. Similarly, if DDP groups are defined, the list is checked before a response to the broadcast message is sent. Refer to "DDP Groups" and "Network Security" in this chapter for additional information.

Unless otherwise specified, a node uses the Automatic Network Configuration mode of MSM, which enables the node to communicate with all other nodes in the network operating in Automatic Configuration mode.

Note Depending on the type of logical or physical link used, you may not be able to use Automatic Network Configuration.

Although this is the normal operating mode, it is not desirable in all cases. For example, assume that a system manager wants to create a network consisting of a single MSM system operating as a server and a large number of personal computers running MSM that access data on the server through the network. In such a configuration, none of the PCs in the network ever need to access data on the other PCs connected to the network.

In Automatic Configuration mode, a Circuit table would be built on each PC that included a circuit entry for the server as well as every PC on the network. This approach would waste memory space and increase the amount of time required to search the table on each PC. For optimum performance, the server should use the Automatic Network Configuration mode, and each PC node should use the Manual Configuration mode. Refer to "DDP System Parameters" in this chapter for additional information on manual mode.

Accessing Globals Through DDP

The DDP feature of MSM allows users to access globals on remote systems. This is done by either specifying an explicit reference to the global or by implicitly referencing the data through the system's Translation and Replication facilities. The following sections describe how nodes in the network are identified and the method used for making explicit and implicit global references.

Identifying a Node

When MSM is started, the first volume group mounted is referred to as the System or Boot volume group. This volume group's name is the name that is automatically assigned to the node (volume group name is the node name). Because data is accessed by a reference that includes a volume group name, all mounted volume groups within the network must have unique names.

When a new volume group is mounted on a node, the system determines whether any other volume group currently mounted has the same name. If so, the system issues a message indicating that a duplicate volume group exists. It then mounts the volume group with the duplicate name. If a global is referenced which resides on a duplicate volume group, MSM takes the first volume group it finds that matches the request, beginning with the system making the request.

Extended Global Notation

MSM's extended global notation allows you to access information stored in different User Class Identifiers (UCIs). The specified UCI may be within the same volume group, a different volume group on the same machine, or a different volume group on another machine.

In all cases, the same method is used to make the global specification. The following examples illustrate the two global reference formats that can be used for an extended reference:

^|*UCIName,VolGroup*|*GlobalVar*

^*[UCIName,VolGroup]GlobalVar*

UCIName is the three-character name of the UCI, and *VolGroup* is the three-character name of the volume group. The first format conforms to the new ANSI Standard specification (uses vertical bars). The second format was used before the ANSI Standard was updated (uses brackets). It has been retained in the system to allow for upward compatibility.

As an alternative, the specified *UCIName* value may include both the UCI name and the volume group name separated by a comma. The specified *UCIName* and *VolGroup* may be on the same system or on a different system. A reference of this type is referred to as an *explicit global reference*.

Internally, the system follows a specific search path to locate the requested information. Refer to "Resolving Global References" in this chapter for additional information on this search path. Extended notation may be used even if DDP is not installed. In this case, references are limited to globals contained on the current system.

The extended global notation described above may be specified as part of the argument that is passed to any of the following commands and functions:

SET	$NEXT
KILL	$ORDER
LOCK	$QUERY
ZALLOCATE	$ZNEXT
ZDEALLOCATE	$ZORDER
$DATA	$ZPREVIOUS
$GET	$ZSORT

The extended global notation can also be used with any other command and function MSM when it is specified as part of an expression.

Translation and Replication

To improve flexibility in a network environment, the MSM system includes a Translation feature and a Replication feature. The Translation feature allows the system manager to specify the precise location of one or more global variables. Whenever a program makes an implicit reference (does not precede the global name with brackets) to a translated global, the system internally supplies the UCI name and the Volume Group name from the associated entry in the Translation table.

Global variables can be moved between volume groups and nodes in the network in a manner transparent to application programs. When MSM-NET is not enabled, this feature still allows global variables to be moved between UCIs and volume groups on the same system.

Replication acts in a manner that is conceptually similar to translation. Rather than using the information in the Replication table to determine the location of a global, it is used to indicate where a duplicate copy of the data is to be maintained. Whenever a global with an entry in the Replication table is updated (a node is added, changed, or deleted), the system duplicates the action on the corresponding globals as indicated by the Replication table.

This duplicate action is performed using the UCI name and Volume Group name specified in the Replication table. The replicated action can take place on the same system or a different system in the network. Refer to "Configuration Utilities" in the *MSM Utility Manual* for additional information on the Translation and Replication features.

Resolving Global References.

When a global is referenced explicitly or implicitly, the system follows a well-defined search path to determine where the requested information is located. The system uses the following search criteria for every global reference:

1. If the global name is preceded by brackets containing a *UCIName* and optionally, a *VolGroup* name, the reference is explicit. No additional searching is required to determine which volume group and UCI contain the global. If a *VolGroup* is not specified, it is assumed that the specified UCI exists within the current volume group. Processing continues with step 4 below.
2. If the global name is not preceded by brackets, the system searches the UCI Translation table to determine if a translated reference is being made. If so, the *UCIName* and *VolGroup* are extracted from the Translation table and no additional searching occurs. Processing continues with step 4 below.
3. If neither of the first two cases is true, the system assumes that the global is in the current UCI on the current volume group. Processing continues with the next step.
4. The system next searches the list of volume groups mounted on the local system, including any mounted Remote Volume Groups. If the system finds the volume group identified in steps 1 through 3 above, the global reference is made in that volume group.
5. If the volume group is not mounted on the local system, the system searches the Circuit tables to determine whether the volume group is mounted on a remote machine. If it is, the appropriate request is made through the network.

If the specified volume group is not found, a <NOSYS> error is generated. If the specified UCI is not found, a <NOUCI> error is generated. If the global node does not exist, it is considered undefined; an <UNDEF> error may result, depending on the operation performed.

Installing MSM-NET

MSM-NET for UNIX systems shares network hardware with other UNIX network facilities. The network adapter cards typically are installed by the manufacturer and are configured to support high-level communication protocols such as TCP/IP, Network File Systems (NFS), and so on. The following sections outline the general requirements necessary for installing the MSM-NET package.

UNIX Software Requirements

Generally, there are no special requirements within the UNIX environment that must be considered when installing the MSM-NET software package. However, on many systems the Ethernet communications controller will not function properly unless the TCP/IP software feature of UNIX has been installed and configured in the system.

If this feature is not installed, then any attempt by MSM-NET to access the communication link (to open a communications path to the controller) will fail. Refer to the appropriate UNIX documentation supplied with the system for additional information on installing and configuring the TCP/IP feature.

AIX Network Driver

The IBM RS/6000 series of computers requires additional network driver software to be installed. The MSM-NET driver for AIX is distributed on a single 3.5-inch floppy diskette. This software is installed on the RS/6000 using the AIX SMIT System Management utility. The following example illustrates how to install the MSM-NET driver in AIX Version 4.2.1.

1. Log on to AIX as **root**.
2. Use the following command to invoke the SMIT utility:

   #smit **RETURN**
3. On the System Management screen of the MSM-NET driver installation program, the default response, **Software Installation and Maintenance,** should be highlighted. If it is not, use the arrow keys to highlight this option and then press **RETURN**.
4. On the Installation and Maintenance screen, highlight the default response, **Install and Update Software**, and then press **RETURN**.
5. On the Install and Update Software screen, highlight the default response, **Install and Update from LATEST Available Software,** and then press **RETURN**.
6. On the next screen, only the INPUT Device/directory for software field initially is displayed. At this field, enter **/dev/rfd0** and press **RETURN**. The system displays the remaining fields on this screen.
7. In the COMMIT software field, press **TAB** to change the value from No to Yes.

8. In the VERIFY software field, press **TAB** and then press **RETURN**. The installation of the MSM-NET software begins.

   If you are upgrading an older version of the MSM-NET driver, AIX detects it and automatically overwrites the older version. Whether installing or upgrading, output similar to the following displays on-screen as part of the MSM-NET installation procedure:

```
indysllp –acgNqQwXd /dev/fd0 –f File 2>&1

File:
     msm.msmnet.obj        4.2.4.1

            Pre-installation Verification…

Verifying selections…done

installp: APPLYing software for:
    msm.msmnet.obj 4.2.4.1

msmnet0 msmnet0 Available
Finished processing all filesets.
```

   After the installation is complete, the system returns to the Install Software with Updates screen.

9. Press **F10** to exit the SMIT utility and return to UNIX. Before using the DDP facility of MSM-UNIX on the RS/6000, use the SYSGEN utility of MSM to configure the software. Refer to the *MSM User's Guide* for information on configuring DDP.

Configuring the MSM-NET Software

Use the SYSGEN utility to define, configure, and maintain all aspects of DDP in MSM. Invoke SYSGEN and then either select an existing configuration or create a new configuration.

On the SYSGEN menu for the selected configuration, select the Network Configuration option. The system responds with the following.

```
Select Option: 15 - Network Configuration

Available Functions:

     1 - DDP System Parameters
     2 - Link Definition
     3 - Circuit Definition
     4 - DDP Groups
     5 - Network Security
     6 - LAT System Parameters
     7 - LAT Node Management
     9 - OMI Translation Table
    10 - Display Network Configuration
    11 - Workstation Server Configuration
    12 - User-Defined Services
    13 - Telnet Service Configuration
    14 - API Service Configuration
    15 - PDQweb Service Configuration

Select Option: ?

    Select option by specifying the option number or supplying enough
    characters to identify the option,
    To get help information specific to an option, enter '?' followed
    by the option number,
    Enter <RETURN> or '^' to exit with no action,
    Enter '^L' to get a list of options.

Select Option:
```

DDP System Parameters

Use this option to specify general system parameters associated with the DDP functions and to tune the DDP system for optimum performance across the network.

DDP buffers are used to send messages between nodes. When a circuit is started, input and output buffers are allocated for its use. If there are not enough free buffers to satisfy the circuit requirements, the circuit is not started. The number of buffers required for a circuit depends on the protocol used between the two nodes. A circuit using MSM protocol requires 10 buffers, one using OMI protocol requires two buffers, and one using DSM protocol requires 50 buffers. If LAT is used, each LAT circuit requires three buffers. An additional one buffer per circuit must be available to handle broadcast messages when using Automatic Configuration.

Circuit entries are allocated during system startup based on the maximum number of circuits specified. If a circuit start request is received when there are no free entries, the circuit cannot be started.

The number of DDP servers that are started determines how well the local system can respond to requests from remote systems. When there are more servers, the local system can respond to more simultaneous requests.

If a request is made to a remote system and no response is received in four seconds, the request is retransmitted. This cycle continues until the maximum response time is exceeded. The request then is terminated with a <DSTDB> error.

With both DDP and Remote Volume Groups, jobs from different systems may access the same global data. To ensure compatibility with applications that are not aware of networks, $JOB values should be unique across the network. If application code expects the $JOB system variable to be unique for each job, the code can fail to function properly when using Remote Volume Groups. One way to work around this problem is to recode the application to use the $SYSTEM variable in combination with $JOB system variable.

MSM provides a DDP parameter, the *job base number*, for use when modifying the application is not practical. This number is added internally to the $JOB value for each job. If more than one client system is using the same server as a Remote Volume Group, each client system can be assigned a unique job base number. As long as the difference between the job base numbers for the clients is greater than the size of the licenses, all job numbers will be unique throughout the network. For example, if the client systems were all 512-user systems or smaller, then setting the job base number to multiples of 1000 ensures unique $JOB values.

Circuits may be built automatically based on broadcast messages or they may be manually specified with the SYSGEN utility. Networking may be started automatically when the MSM system is started or may be controlled manually with the DDP utility.

When the DDP server receives a request from another system to start or stop a circuit, it can write a message to the system console about the circuit's new status. Messages also can be generated for various error conditions detected on the links. If these messages interfere with other operations on the console terminal, they may be disabled.

```
Select Option: 1 - DDP System Parameters

Number of DDP Buffers <10>: ?

    Enter the number of DDP buffers to be allocated by the system.
    Buffers are used to send messages between nodes.  If the number is
    too small, circuits cannot be started.

    Each DDP/RVG circuit requires 10 buffers.
    Each LAT circuit requires 4 buffers.

Number of DDP Buffers <10>: 12

Maximum Number of DDP Circuits <6>: ?

    Enter the number of entries to be built in the Circuit Table. One
    entry is required for each Circuit that can be accessed.

Maximum Number of DDP Circuits <6>: 8

Number of DDP Servers <1>: ?

    Enter the number of servers that will be started by the system when
    DDP is enabled. If more than one server is specified, multiple
    global references can be processed concurrently.

Number of DDP Servers <1>: 2
```

```
Number of Seconds to Wait For Network Response <30>: ?

    Enter the number of seconds that the system will wait to receive a
    response to a message that is sent across the network. If a
    response is not received within the specified time, the system
    marks the node unreachable and terminates the request with a
    <DSTDB> error.

Number of Seconds to Wait For Network Response <30>: 60

Job Number Base for This System <0>: ?

    Enter a number that will be the base for this system. Every job
    created on this system will have this base number added to its job
    number. The value should be chosen to provide unique job numbers
    across the network. This is not required by the system, but ensures
    integrity when the job number ($JOB) may be used as a global
    subscript remotely mounted volume groups.

Job Number Base for This System <0>: 1000

Automatic Network Configuration <Y>: ?

    Indicate whether Automatic Network Configuration is to be enabled
    for this node. If yes, then the node builds the Circuit Table from
    broadcast messages received from other nodes. If no, then the
    system manager must specify the Circuit information for each remote
    node that is to be accessible from this node.

Automatic Network Configuration <Y>: N

Startup DDP During System Startup <N>: ?

    Enter YES to allow the startup utilities to automatically start DDP
    as part of the normal system startup.
    Enter NO to control DDP startup manually using the DDP utility.

Startup DDP During System Startup <N>: Y

Display DDP Messages <Y>: ?

    Enter YES to display DDP server messages to the system console.
    Enter NO if DDP server messages are not to be displayed.

Select Option:
```

Link Definition

Use the Link Definition option to define the type of network link that will be used to communicate between systems. The information required by MSM-NET to access the link depends on the type of link. The following sections describe how to configure each link type supported by MSM-UNIX.

Not all of the link types described below are supported on all MSM-UNIX platforms. The Ethernet link is available on all of the platforms listed below. UDP/IP also is available on most platforms. The TCP/IP link is functional for only a few platforms. Contact Micronetics Design Corporation or your reseller for current information on support features for your system.

From time to time, the list of UNIX platforms that support various MSM-NET protocols is expanded. When this occurs, the README.TXT file contained on the distribution media includes documentation about the new systems that are supported. General information is provided about the feature, the adapter name, restrictions, and so on.

```
Select Option: 2 - Link Definition

Select Link Number: ?

    Enter the number of the Link to edit, or a new number to add a
    Link.  Or, enter an existing Link number preceded by a minus sign
    ('-') to delete a Link from the table.

Select Link Number: 1

Link Type:

    1 - Ethernet
    2 - TCP/IP
    3 - UDP/IP

Select Link Type <1>: ?

    Enter the index number for the type of interface that this link
    will use.

        1 - Ethernet
        2 - TCP/IP
        3 - UDP/IP

Select Link Type <1>:
```

Ethernet links

MSM-UNIX supports the IEEE 802.3 standard for Ethernet communication. Links in MSM that adhere to this standard are referred to as *Ethernet links*. Each node connected to an Ethernet link includes an Ethernet controller and an Ethernet transceiver. The physical connection between nodes is accomplished by connecting the transceivers together using coaxial cable.

In some cases, the transceiver is built into the controller, eliminating the need for a separate external device. Using newer technology, the coaxial cable may be replaced with twisted pair telephone wire. The twisted pair wire may be connected to the coaxial cable using an external controller to bridge the two networks. All of these network types are variations on the standard Ethernet network.

By design, every Ethernet controller manufactured has an unique address. The address consists of 12 hexadecimal digits and is typically shown as six groups of two digit numbers separated by hyphens (for example: AB-CD-EF-12-34-56).

When a message is sent to a node, the Ethernet address of the controller on the remote system is used to specify the location. With the exception of broadcast and multicast messages, a controller only receives messages that contain its unique controller address. Every message received by a node includes the controller address of the node that sent the message.

Ethernet adapter name

In order for MSM-NET to access an Ethernet communications controller, it must know the device name within UNIX that is associated with the adapter. Although there is no published standard for how Ethernet devices are named within UNIX, common naming conventions are followed by many hardware manufacturers.

For example, in most systems, the Ethernet controller is identified by its physical device name (for example: /dev/ether0, /dev/net0). In other systems, the Ethernet communications controller is identified by a logical device name (for example: ln0). The appropriate device name must be specified to the MSM-NET package when a communication link is defined.

The following table lists the Ethernet adapter names that are used for each UNIX system supported by the MSM-NET feature.

UNIX Ethernet Adapter Names

Model Name	Ethernet Adapter Name
DG AViiON Series	/dev/hken0
Digital UNIX	ln0
Hewlett Packard	/dev/ether0
IBM RS/6000 Series	/dev/ent0
SCO UNIX	/dev/net0
Sequent S Series	/dev/enet/eg0
Sun Solaris	/dev/nit

```
Select Link Number: 1

Link Type:

    1 - Ethernet
    2 - TCP/IP
    3 - UDP/IP

Select Link Type <1>: 1

UNIX Ethernet Adapter: ?

    Enter the Ethernet device or adapter name for the link. For
    example, /dev/ether0, /dev/en1, 1n0, etc.

UNIX Ethernet Adapter:  /dev/ether0

Current Links Defined:

Link   Device Name
----   -----------
  1    /dev/ether0

Select Link Number:
```

Native Token Ring links

MSM-UNIX, when running under AIX, supports the IEEE 802.5 standard for Token Ring communication using IEEE 802.2 packets. Links in MSM that adhere to this standard are referred to as Token Ring links and operate in native mode. As the name implies, Token Ring networks use a token-passing access method and a ring topology that resembles a star. Commonly, computers are connected to the network with shielded or unshielded twisted pair cable to a wiring concentrator called a Multistation Access Unit (MAU). The networks typically operate at either 4 MB per second or 16 MB per second. Computers usually must be located within 100 meters of the MAU. A maximum of 72 or 260 stations can be connected depending on whether unshielded or shielded cable is used. Each MAU supports up to eight stations and concentrators can be linked to form extended rings.

Conceptually, a Token Ring network is based on the use of a single token that is passed serially around the ring. When a station wishes to transmit, it must wait until it receives the token. The receiving station then changes the token from free to busy and immediately transmits data following the busy token. When the station's interval expires or it finishes sending data, it transmits a free token to indicate the network is available.

Because Token Ring networks are currently supported only for the AIX operating system, there is no separate SYSGEN option for it. To define a Token Ring link, select the Ethernet Link option and enter the device name of the Token Ring adapter, (for example: /dev/token0). Although the MSM utilities will refer to the link as an Ethernet link, it will operate internally as a Token Ring link.

TCP/IP links

Transmission Control Protocol/Internet Protocol (TCP/IP provides a high-level interface that allows MSM-UNIX to share the physical hardware controller with other applications running in the UNIX system. TCP/IP links do not support broadcast or multi-cast messages. The address of the remote system must be known to the local system. Each application that shares the controller must also be assigned a unique port number that allows it to receive messages.

TCP/IP links only support OMI protocol. No other type of M to M communication is possible through a TCP/IP link. TCP/IP is the only type of link that can be used by OMI. OMI circuits must be manually configured.

TCP/IP links are NOT required for socket communications via device 56.

In order for MSM-NET to communicate with other systems via TCP/IP, the port number of the remote system must be known. A *port number* is a unique number that identifies MSM from all other application software on a system. The port number is a number from 0 through 65535, although 0 through 1023 are reserved. The actual value has no particular significance, except that it must be unique for each and every application (for this purpose, MSM is considered an application) using TCP/IP or UDP/IP on the same machine.

```
Select Link Number: 2

Link Type:

    1 - Ethernet
    2 - TCP/IP
    3 - UDP/IP

Select Link Type <1>: 2

Port Number <1234> ?

    Enter TCP/IP port number (in decimal) that this link will use for
    listening.  The default value for MSM systems is 1234.

Port Number <1234> RETURN

Current Links Defined:

Link   Device Name
----   -----------
  1    /dev/ether0
  2    TCP/IP        Port Number:  1234

Select Link Number:
```

UDP/IP links

Universal Datagram Protocol/Internet Protocol (UDP/IP) provides an alternative to TCP/IP communications and gives MSM more control over the types of messages that can be sent. DDP and RVG packets can be sent via UDP/IP links. If all MSM systems in the network use the same port number, then it is possible to support Automatic Network Configuration through the broadcast or multi-cast messages that MSM sends when DDP starts up. It also is possible to manually configure circuits over UDP/IP links using a unique port number for each system.

In order for MSM-NET to communicate with other systems via UDP/IP, the port number of the remote system must be known. The port number is a number from 0 through 65535, although 0 through 1023 are reserved. The value must be unique for each and every application (for this purpose, MSM is considered an application) using TCP/IP or UDP/IP on the same machine.

TCP/IP and UDP/IP use the same port numbers. There cannot be two applications on the same system, one using TCP/IP and the other using UDP/IP, which both use the same port number.

Unlike OMI circuits using TCP/IP links, MSM circuits using UDP/IP links can be automatically configured. When DDP starts up in Automatic Configuration Mode, MSM broadcasts a message to the network using its own UDP/IP port number. If all the other MSM systems using UDP/IP use the same port number, they will see the broadcast message and recognize the new system. In response, they will send their own address and system information to the broadcasting system.

```
Select Link Type <1>: 3 - UDP/IP

Link Type:

     1 - MSM Data Link
     2 - TCP/IP
     3 - UDP/IP
     4 - IPX

Select Link Type <3>: RETURN

IP Port Number <33087> RETURN

Network Mask <0.0.0.0> RETURN

Bind Address <0.0.0.0> ?

    Enter a host IP address to bind to for the UDP/IP link. An address
    of 0.0.0.0 means MSM will bind to the default address for this
    system.  If this system has more than one IP address, specify which
    address to associate with this particular UDP/IP link. The address
    must be four numbers, separated by periods. Each number must be
    less than 256.

Bind Address <0.0.0.0> RETURN

Current Links Defined:

Link   Device Name   Port Number   Network Mask   IP Address
----   -----------   -----------   ------------   ----------
  1    0001
  2    IPX           33087
  3    UDP/IP        33087         0.0.0.0        0.0.0.0

Select Link Number:
```

Circuit Definition

When a node is set up to use the Manual Configuration mode of MSM, use this SYSGEN option to define each circuit within a network that can be accessed. Only circuits that are explicitly defined in this manner can be accessed. OMI and ISM circuits must be manually configured.

When circuits are configured manually, you must specify a list of volume groups that reside on the remote systems. Specific additional information is required for each circuit type. The following sections describe the manual configuration process for all types of circuits supported by MSM-UNIX.

Ethernet circuits

When manually configuring MSM systems, DSM systems, and ISM systems on an Ethernet link, the Ethernet address and network timeout must be specified.

```
Select Option: 3 - Circuit Definition

Current Circuits Defined:

Cir  Status   Link  Volume Groups  Remote System      Mode   Defn
---  -------  ----  -------------  -----------------  -----  ----
VAX  Ena,Rea    1   VAX,CUS        12-34-56-78-9A-BC  DSMV2  Man

Select Circuit: ?

    Enter the three-character name of the remote node that is to be
    defined.  Or, you may precede the name with a minus sign (-) to
    delete the node.

Select Circuit: TST

Communications Mode:

    1 - Micronetics Standard MUMPS (MSM)
    2 - Digital Standard MUMPS (DSM)
    3 - Open MUMPS Interconnect (OMI)
    4 - Intersystem Standard MUMPS (ISM), VAX
    5 - Intersystem Standard MUMPS (ISM), PDP11

Select Communications Mode <1>: 1

DDP Link Number <1>: ?

    Enter the number of the communications Link that is to be used to
    access the specified node.

DDP Link Number <1>: 1

Mounted Volume Groups <TST>: ?

    Enter one or more three character names, separated by commas, to
    indicate the volume groups that are mounted on the node. A maximum
    of 8 mounted volume groups may be specified.

Mounted Volume Groups <TST>: TST,BPS,JWC

Ethernet Physical Address: ?

    Enter the physical address of the Ethernet Controller on the
    remote node.  It must be in the form of six sets of two-digit
    hexadecimal numbers, separated by hyphens (-).

Ethernet Physical Address: FE-DC-BA-98-76-54
```

```
Network Response Timeout:  ?

    Enter the timeout value, in seconds, that is to be used for  this
    circuit.  This value overrides the default set under DDP System
    Parameters.  The minimum allowable value is 16, and is internally
    rounded-up to a multiple of 4 seconds.

Network Response Timeout:  20

Current Circuits Defined:

Cir  Status   Link  Volume Groups  Remote System      Mode   Defn
---  -------  ----  -------------  -----------------  -----  ----
VAX  Ena,Rea    1   VAX,CUS        12-34-56-78-9A-BC  DSMV2  Man
TST  Ena,Rea    1   TST,BPS,JWC    FE-DC-BA-98-76-54  MSMV3  Man

Select Circuit Number: RETURN

Select Option:
```

OMI circuits

When configuring OMI circuits on a TCP/IP link, specify the circuit type, communications link number, and remote server information. Since OMI circuits can process transactions in only one direction, circuit types are defined by their function as message receivers or senders. Two circuits, a server and an agent, must be defined when two-way transactions are desired.

A *server* receives data requests directed to the local system from remote systems. Because requests can be received from any remote system, the server's name is not important. An *agent* directs data requests from the local system to remote systems and must use the same name as the remote system.

When you define an agent, specify the communications link number and information about the remote system to be accessed. In addition to the remote system name, enter the remote system's Internet Protocol Address and Port Number. This information node can be saved in the database so that changes made to the in-memory circuit table are loaded each time the system is started up.

The concepts of UCIs and volume groups do not exist in the OMI specification. The equivalent term for an application database in OMI is Environment. Because MSM operates only with volume groups, a mechanism exists for translating a UCI,VOL pair to an OMI Environment Name. When defining an OMI circuit, a list of "virtual" volume groups is entered which are actually entries in the OMI Translation table.

```
Select Option: 3 - Circuit Definition

Current Circuits Defined:

Cir  Status   Link  Volume Groups  Remote System      Mode   Defn
---  -------  ----  -------------  -----------------  -----  ----
VAX  Ena,Rea    1   VAX,CUS        12-34-56-78-9A-BC  DSMV2  Man
TST  Ena,Rea    1   TST,BPS,JWC    FE-DC-BA-98-76-54  MSMV3  Man

Select Circuit: FOR

Communications Mode:

     1 - Micronetics Standard MUMPS (MSM or MUMPS/VM)
     2 - Digital Standard MUMPS (DSM)
     3 - Open MUMPS Interconnect (OMI)
     4 - Intersystem Standard MUMPS (ISM), VAX
     5 - Intersystem Standard MUMPS (ISM), PDP11
     6 - MSM Version 2 or Version 3 (MSMV2)

Select Communications Mode <1>: 3
```

```
Is this circuit for a Server or an Agent <A>: ?

    An OMI circuit can handle DDP transactions in only one direction.
    A Server circuit receives requests from other systems that want to
    access data that exists on this system. An Agent circuit sends
    requests from jobs on this system that want to access data that
    exists on another system (it acts as the agent for the jobs).

    If requests will be made in both directions, there must be two
    circuits defined: one agent and one server.  Note that the name of
    an agent circuit should match the system name of the remote system.
    The name of a server circuit is not significant because it may
    handle requests from any remote system.

    Enter 'S' for server, or 'A' for agent.

Is this circuit for a Server or an Agent <A>: AGENT

DDP Link Number <1>: ?

    Enter the number of the communications Link that is to be used to
    access the specified node.
    Enter ^L to list the currently defined links.

DDP Link Number <1>: ^L

Current Links Defined:

Link   Device Name
----   -----------
  1    /dev/ether0
  2    TCP/IP        Port Number:  1234
  3    UDP/IP        Port Number:  33087

DDP Link Number <1>: 2

Mounted Volume Groups <FOR>: RETURN

Remote Server Name: ?

    Enter the server name that will be sent to the remote system when
    the circuit is started. The use of this field is defined by the
    remote system. The maximum length of the name is 15 characters.
    Enter '-' to change an existing value to the null string.

Remote Server Name: SERVERA

IP Address: ?

    Enter the Internet Protocol address of the remote system. It must
    be in the form of four sets of numbers less than 255, separated by
    dots (for example, 123.17.20.9).

IP Address: 130.56.1.15

IP Port Number: ?

    Enter the Internet Protocol port number that the remote system uses
    for listening.

IP Port Number: 3346

Save this node into the database <Y/N>: ?

    Circuit table changes have already been made to the in-memory
    circuit table.  Answer Y if you wish to have this modification
    saved in ^SYS and loaded each time the system starts up.

Save this node into the database <Y/N>: N
```

```
Current Circuits Defined:

Cir  Status   Link  Volume Groups  Remote System      Mode   Defn
---  -------  ----  -------------  -----------------  -----  ----
VAX  Ena,Rea    1   VAX,CUS        12-34-56-78-9A-BC  DSMV2  Man
TST  Ena,Rea    1   TST,BPS,JWC    FE-DC-BA-98-76-54  MSMV3  Man
FOR  Ena,Rea    2   FOR            130.56.1.13~3346   OMIV1  Man

Select Circuit Number: RETURN

Select Option
```

MSM circuits via UDP/IP

Universal Datagram Protocol/Internet Protocol (UDP/IP) links enable communication between MSM systems only. When configuring circuits to MSM systems on a UDP/IP link, specify the mounted volume group name(s) and the IP address and port number.

```
Select Option: 3 - Circuit Definition

Current Circuits Defined:

Cir  Status   Link  Volume Groups  Remote System      Mode   Defn
---  -------  ----  -------------  -----------------  -----  ----
VAX  Ena,Rea    1   VAX,CUS        12-34-56-78-9A-BC  DSMV2  Man
TST  Ena,Rea    1   TST,BPS,JWC    FE-DC-BA-98-76-54  MSMV3  Man
FOR  Ena,Rea    2   FOR            130.56.1.13~3346   OMIV1  Man

Select Circuit: OPN

Communications Mode:

     1 - Micronetics Standard MUMPS (MSM or MUMPS/VM)
     2 - Digital Standard MUMPS (DSM)
     3 - Open MUMPS Interconnect (OMI)
     4 - Intersystem Standard MUMPS (ISM), VAX
     5 - Intersystem Standard MUMPS (ISM), PDP11
     6 - MSM Version 2 or Version 3 (MSMV2)

Select Communications Mode <1>: 1

DDP Link Number <1>: ^L

Current Links Defined:

Link   Device Name
----   -----------
  1    /dev/ether0
  2    TCP/IP        Port Number:  1234
  3    UDP/IP        Port Number:  33087

DDP Link Number <1>: 3

Mounted Volume Groups <OPN>: OPN,VAR

IP Address: ?

    Enter the Internet Protocol address of the remote system. It must
    be in the form of four sets of numbers less than 255, separated by
    dots (for example, 123.17.20.9).

IP Address: 130.56.1.32

IP Port Number <33087>: ?

    Enter the Internet Protocol port number that the remote system uses
    for listening.

IP Port Number <33087>: 33087
```

```
Network Response Timeout <60>: ?

    Enter the timeout value, in seconds, that is to be used for this
    circuit. This value overrides the default set under DDP System
    Parameters. The minimum allowable value is 16.

Network Response Timeout <60>: 60

Save this node into the database<Y/N>: N

Current Circuits Defined:

Cir  Status   Link  Volume Groups  Remote System      Mode   Defn
---  -------  ----  -------------  -----------------  -----  ----
VAX  Ena,Rea    1   VAX,CUS        12-34-56-78-9A-BC  DSMV2  Man
TST  Ena,Rea    1   TST,BPS,JWC    FE-DC-BA-98-76-54  MSMV3  Man
FOR  Ena,Rea    2   FOR            130.56.1.13~3346   OMIV1  Man
OPN  Ena,Rea    3   OPN,VAR        130.56.1.32~33087  MSMV3  Man

Select Circuit Number: RETURN

Select Option
```

DDP Groups

In MSM-NET, nodes are organized into related groups. Two nodes within the same group may communicate with each other. Communication is not permitted between two nodes that are not members of the same group.

A node may belong to more than one group. MSM's standard global protection and DDP groups features are used to ensure security. Refer to the *MSM User's Guide* for additional information on the security feature.

```
Select Option: 4 - DDP Groups

Enter DDP Group Number: ?

    Enter a list of DDP group numbers from 0 to 15 to which this system
    will be added as a member. Each system can belong to more than one
    DDP group.  This system will be allowed to perform DDP transactions
    to all systems in all groups of which it is a member, though access
    can be further restricted via Network Security passwords. If no
    groups are specified, then this system automatically is assigned
    membership in DDP group 0.

    Enter a list of group numbers, separated by commas, to add
    membership to the groups.
    Enter a '-' in front of the list to remove membership from the
    listed groups.
    Each part of the list may be a range in the form num1-num2.
    Enter ^L for a list of groups to which this system currently
    belongs.
    Enter '^' or RETURN to return to the DDP Menu.

Enter DDP Group Number: ^L

    No groups specified, default membership is group 0.

Enter DDP Group Number: 2  ...membership to group 2 added.

Enter DDP Group Number: 5  ...membership to group 5 added.

Enter DDP Group Number:
```

Network Security

In MSM-NET, you can assign each node a password. When a logical connection is attempted between two nodes, the system checks the passwords. If they do not match, the logical connection is not established and no communication is permitted between the nodes.

MSM's standard global protection feature and network passwords are used to ensure network security. Refer to "Globals" in the *MSM User's Guide* for additional information on security features.

Network password

Use this option to set the password for the current node. Passwords are required when the Security Challenge feature of MSM is enabled. Refer to the following section for information on the Security Challenge function.

```
Select Option: 1 - Network Password

Enter Network Password: ?

    Enter a Network Security password. This password will be broadcast
    to all other systems on the network when DDP is enabled. Other
    systems which use Security Challenge must have this password in
    their database in order to allow the connection. Systems which do
    not use Security Challenge will allow connection regardless of
    whether this system has a password.

    Enter '-' to delete a password and disable Security Challenge.
    Enter '^' to return to the Network Security Menu.

Enter Network Password: PASSWD

Select Option:
```

Security challenge

Use this option to enable MSM's Security Challenge feature, which affects both MSM and DSM systems. In order for communication to occur, valid passwords must be defined for each system.

```
Select Option: 2 - Security Challenge

Enable Security Challenge? <N>: ?

    If Security Challenge is enabled, only systems with a network
    password will be allowed to connect to this system. Otherwise, all
    systems in the same DDP group will be allowed access to this system
    over the network.

    Enter YES to enable Security Challenge. This utility will then
    solicit the names and passwords of all systems that will be allowed
    to connect to this one.  When a remote system attempts to gain
    access for DDP, its password must have been entered on this system
    via the Security Challenge Option. Systems which have no network
    password will not be allowed access if Security Challenge is
    enabled.

    Enter '^' to return to the Network Security Menu.

Enable Security Challenge? <N>: Y

Enter System Name: ?

    Enter the three-letter name of the system (node) to be allowed
    network access.
    Enter -name to delete a previously entered system name from the
    database.
    Enter ^L for a list of system names already entered.
    Enter '^' to backup to the previous question.
```

```
Enter System Name: VAH

Enter Network Password for VAH: ?

    Enter the network password for the remote system. This password
    must have been entered through the Network Security option in
    SYSGEN (for MSM systems), or through the Circuit Connection
    Security facility for DSM systems.
    Enter '^' to backup to the previous question.

Enter Network Password for VAH: VAHPSWD

Enter System Name: MDC

Enter Network Password for VAH: MDCPSWD

Enter System Name: RETURN
```

OMI Translation Table

Open M Interconnect (OMI) is a network communications protocol established by the M Development Committee (MDC). OMI allows different implementations of M to share data across the network. This protocol must be used when defining a DDP circuit that is to connect with an M implementation not directly supported by MSM.

In the MSM system, all global references are resolved to a UCI and volume group that specify the location of the global. Refer to “Resolving Global References” in this chapter for a description of how the UCI and volume group are determined for a given global reference.

In other M implementations, the location of a global may be specified other than by UCI and volume group. There must be a method of translating the UCI and volume group of an MSM reference to a location that the other system recognizes.

In OMI, the location of a global is called an *environment*. Each M implementation defines the environments that it can recognize. For example, if another system sends a global request to the MSM system, it must specify an environment of ‘UCI,VGP’ which names the UCI and volume group where the global exists.

In the MSM system, if a global reference resolves to a volume group that is defined as belonging to an OMI circuit, the system searches the OMI Translation table. If an entry is found that matches both the UCI and the volume group of the global reference, the corresponding environment name is passed to the other system to specify the location (on the other system) of the requested global. If a matching entry is not found in the Translation table, an environment of ‘UCI,VGP’ is sent to the other system. If the other system does not recognize the environment that is passed to it, an error is indicated to the MSM system.

```
Select Option: 9 - OMI Translation Table

Enter OMI Translation Index: ^L

Index    UCI,VGP    OMI Environment Name
-----    -------    --------------------
  1      FKI,RMT    SYSTEMA

 Enter OMI Translation Index: ?

    Enter an existing index number to edit an entry in the OMI
    Translation Table, or enter a new index number to add an entry to
    the table.
    Enter -index to delete an entry from the table.
    Enter ^L to display the current contents of the table.
    Enter ^ to return to the menu.

Enter OMI Translation Index: 2

Enter UCI,VGP to Translate: ?

    Enter a UCI name and Volume Group in the form UCI,VGP. Any global
    references which resolve to this UCI and volume group will be
    translated to the OMI Environment Name specified via the next
    prompt.
    Enter ^ to return to the previous question.

Enter UCI,VGP to Translate: VAL,FOR

Enter OMI Environment Name: ?

    Enter an OMI Environment Name to which the UCI,VGP will be
    translated.  When a global is referenced in the UCI specified in
    this entry, the request will be sent via OMI to the specified
    Environment Name.
    Enter ^ to return to the previous question.

Enter OMI Environment Name: SYSTEMB

Enter OMI Translation Index: ^L

Index   UCI,VGP   OMI Environment Name
-----   -------   --------------------
  1     FKI,RMT   SYSTEMA
  2     VAL,FOR   SYSTEMB
```

Display Network Configuration

Use this option to display all current network configuration settings. Enter the device name to which this data is to be output.

```
Select Option: 10 - Display Network Configuration

Enter output device <105>: RETURN

   26-MAY-98    4:20 PM

 - DDP System Parameters -

Number of DDP Buffers..............: 200
Maximum Number of DDP Circuits......: 20
Number of DDP Servers..............: 1
Network Response Timeout............: 60
Job Number Base for This System.....: 0
Automatic Network Configuration.....: Y
Startup DDP During System Startup...: Y
Display DDP Messages...............: N

 - Network Links -

Link LinkTyp  Status  DEVerr RECerr XMTerr BUFerr Local System
---- -------  ------  ------ ------ ------ ------ -----------------
  1  Ethernet On,Ena    0      0      0      0    12-34-56-78-9A-BC

DDP Groups.........................: 0-2

Network Password...................: <Null>
Enable Security Challenge?.........: N

 - Node Name & Description -

Node Name..........................: MDCSYS
Node Description...................: Production system

LAT Groups.........................: 0-2

 - OMI Translation Table -

Table is empty...
```

Custom DDP Security

MSM provides a feature which the system manager can use to develop site-specific DDP and RVG security. This feature is implemented via the VGSECU and DTSECU utilities.

VGSECU and DTSECU are not distributed with the system. They are written by the system administrator. At the appropriate time, the MSM system checks to see if the routine exists in the Manager's UCI. If it exists, it is presumed to have the correct entry point, and the entry point is called. If the routine does not exist, then no security checking is performed.

VGSECU is called when a DDP circuit is being created. The VGSECU routine determines which local volume groups can be accessed by the circuit being created.

The VGSECU utility is called by MSM as an extrinsic function at the label VGLIST, using a command similar to the following:

```
S LIST=$$VGLIST^VGSECU(PROTO,SYSN,ADDR,PASSWD,LICSER,LICFT)
```

The parameters for the function are described in the following table.

Parameter	Description
PROTO	Contains the circuit protocol type. Possible values are "MSMV3" for MSM-Workstation and MSM-Server Version 4.0 or higher, or "DSMV2" for older versions of MSM-Server and non-MSM systems.
SYSN	Contains the system name of the remote system. For an MSM-Workstation connection, this is the null string.
ADDR	Contains the network address of the remote system. For an Ethernet link, this is the 6-byte Ethernet address. For a UDP/IP or TCP/IP link, this is the 4-byte IP address and a 2-byte dynamically assigned port number. For an IPX or SPX link, this is the 4-byte network address, the 6-byte node address, and a 2-byte dynamically assigned port number.
PASSWD	Contains the network password sent by the remote system. For an MSM-Workstation connection, this is the null string.
LICSER	Contains the serial number from the license of the remote system. This parameter is undefined for the DSMV2 protocol.
LICFT	Contains license features of the remote system. Bit 0 indicates a run-time license. Bit 1 indicates a demo license. This parameter is undefined for the DSMV2 protocol.

The return value from the function is a list of mounted volume groups which may be accessed by the remote system. Only volume groups that are in the list will appear in the remote system's circuit table and will be eligible for RVG mounting. If the list is equal to the null string, then the local system will have a circuit to the remote system, but the remote system will not be able to access any volume groups on the local system. If the list is equal to -1, then the circuit is rejected completely.

DTSECU is called when an MSM-Workstation built application requests an RVG mount. The DTSECU routine determines which local volume groups can be mounted by the application. If both VGSECU and DTSECU are present, VGSECU is also called during an MSM-Workstation mount request, but it is the DTSECU routine which determines whether the mount is allowed.

The DTSECU utility is called by MSM as an extrinsic function at the label VGLIST, using a command similar to the following:

```
S LIST=$$VGLIST^DTSECU(ID)
```

The ID parameter contains the Product ID of the application which is requesting an RVG mount. Starting with MSM-Workstation Version 2.0, every application receives a 16-character Product ID when it is built. Applications built with MSM-Workstation Version 1.0 do not have a Product ID, and pass the string “This is a test security string” as the ID.

The return value from the function is a list of mounted volume groups which may be accessed by the remote system. Only volume groups that are in the list will be eligible for mounting.

The following sample code illustrates how to use the VGSECU utility.

```
VGSECU  ;Custom DDP security routine
VGLIST(PROTO,SYSN,ADDR,PASSWD,LICSER,LICFT)  ; Entry point
    ;
    IF $ZBOOLEAN($G(LICFT),#2,1) Q -1 ;disallow circuit to demo license
    ;
    N VG D GETVG^%VGUTIL  ; Get list of all mounted volume groups
    N N,LIST S N="",LIST=""
    F  S N=$O(VG(N)) Q:N=""  I N?3U S LIST=LIST_","_N
    S $E(LIST)=""
    ;
    I SYSN="XXX" Q LIST ; all VGs can be accessed by remote system XXX
    I ADDR=$C(#24,#AF,#8B,0,#12,#C0) Q LIST ; allow complete access by
           ; the system at Ethernet address 24-AF-8B-00-12-C0
    I $G(LICSER)=1234 Q LIST ; allow complete access for Serial # 1234
    I SYSN="AAA" Q "AAA,BBB" ; only AAA and BBB will be accessible
    I $E(SYSN,1,2)="ZY" Q "" ; allow one-way circuit
    Q -1 ; reject circuit
```

The following sample code illustrates how to use the DTSECU utility. This example allows the application with Product ID 12UK-MSMW-SMSM-NH1G to mount only a volume group called WSD. No other applications will be allowed to mount anything.

```
DTSECU  ;Control Workstation application RVG access
VGLIST(ID) ; Entry point
    IF ID="12UKMSMWSMSMNH1G" QUIT "WSD"
    QUIT ""
```

The DDP Utility

The Distributed Data Processing (DDP) feature of MSM includes a system management utility (the DDP utility) that is used to control a node's access to the network (startup and shutdown), to reload Circuit tables, to verify operation of the network, and to perform maintenance functions at the link and circuit level. The following sections describe the capabilities of this utility.

Invoke the DDP Utility

To access the DDP utility, you must be logged on to the Manager's UCI in programmer mode. When the DDP utility is invoked, the system displays the menu shown in the following sample terminal session.

```
>D ^DDP

        MSM Distributed Data Processing Utility

Available Options:

    1 - Start up DDP
    2 - Shut down DDP
    3 - Update Circuit Table
    4 - Verify Circuit Communication
    5 - Link Management
    6 - Circuit Management

Select Option:
```

Start Up DDP

When you use this option to start networking on the machine, all DDP links that are defined and marked Online are started. Refer to "Link Management" in this chapter for additional information on marking links Online and Offline.

```
Select Option: 1 - Start up DDP

How Many Servers Do You Want To Invoke <1>: ?

    Enter the number of DDP servers to be started. Busy systems may
    require more than one server in order to maintain optimum
    throughput on the network.

How Many Servers Do You Want To Invoke <1>: RETURN

DDP Startup ... Link 1 is now enabled

Select Option:
```

Shut Down DDP

When you use this option to stop networking on the machine, all active Links are stopped. As part of the shutdown process, the system broadcasts a message to all other nodes on the network as each Link is stopped.

```
Select Option: 2 - Shut Down DDP

Ok to shut down DDP <N>? Y

DDP Shutdown ... Link 1 is now disabled

Select Option:
```

Update Circuit Table

Use this option to request an Automatic Reconfiguration of the network. The system issues a broadcast message to all systems currently attached to the network requesting that they transmit their node name and currently mounted volume groups. This information is used to update the system's circuit table.

```
Select Option: Update Circuit Table

Broadcast Sent to All Systems

Circuit Table Update Complete

Select Option:
```

Verify Circuit Communication

Use this option to verify that one or more circuits within the network are communicating properly. If more than one circuit is defined, enter the name of the circuit to be verified. Enter the three-character name of a circuit or ALL to indicate all circuits.

```
Select Option: Verify Node Communication

Select Circuit: ?

    Enter the name of the Circuit that is to be tested, or
    Enter '*' to verify all circuits that are active, or
    Enter '^L' to display a list of the active circuits.

Select Circuit: TST

TST ... Circuit Functioning Normally
```

Link Management

Use this option to control functions associated with the communications link and to display link status information.

```
Select Option: 5 - Link Management

Available Options:

    1 - Status of Link
    2 - Enable Link
    3 - Disable Link
    4 - Change Link Status

Select Option:
```

Status of link

Use this option to display the status of all defined links. It includes Link Number, Controller Type, Link Status (Online, Offline, Enabled, or Disabled), Error Counters (device errors, receive errors, transmit errors, and buffer errors), and the Ethernet Address.

```
Select Option: 1 - Status of Link

Link LinkTyp Status DEVerr RECerr XMTerr BUFerr Local System
---- ------- ------ ------ ------ ------ ------ -----------------
  0  ether0  On,Ena    0      0      0      0   12-34-56-78-9A-BC
```

Enable link

Use this option to start a specified Link in the network. Enter the name of the link to be started. If the link is already operational, the system responds with an appropriate message, and no action takes place.

```
Select Option: Enable Link

Select Link: ?

    Enter the number of the Link that is to be enabled, or
    Enter '*' to enable all links that are defined, or
    Enter '^L' for a list of the defined Links.

Select Link: 0

Link #0 is Now Enabled

Select Option:
```

Disable link

Use this option to stop a specified link in the network. Enter the name of the link to be stopped. If the link is not operational, the system responds with an appropriate message and no action takes place.

```
Select Option: Disable Link

Select Link: ?

    Enter the number of the Link that is to be disabled, or
    Enter '*' to disable all links that are defined, or
    Enter '^L' for a list of the defined Links.

Select Link: 0

Link #0 is Now Disabled

Select Option:
```

Change link status

Use this option to modify the status of a specified link in the network. Enter the name of the link to be modified. You can place the link in either Online or Offline status. When a link is Offline, it cannot be started.

```
Select Option: 4 - Change Link Status

Select Link: ?

    Enter the number of the Link that is to be changed, or
    Enter '^L' for a list of the defined links

Select Link: 1

Link is Currently Online

Available Options:

    1 - Online
    2 - Offline

Select Option: 2 - Offline

Select Option:
```

Circuit Management

Use this option to control functions for individual communications circuits. You can display circuit status and activity, enable and disable individual circuits, reset the statistical counters associated with a circuit, and save the circuit definition for use by the system in Manual Configuration mode.

```
Select Option: 6 - Circuit Management

Available Options:

    1 - Status of Circuit
    2 - Circuit Activity Display
    3 - Enable Circuit
    4 - Disable Circuit
    5 - Reset Statistical Counters
    6 - Save Circuit Information

Select Option:
```

Status of circuit

Use this option to display the status of all defined circuits. The following information is provided: Circuit Name, Circuit Status (Enabled, Disabled, Reachable, or Unreachable), Link Number, Link Status (Enabled, Disabled, Online, Offline), Mounted Volume Groups, Ethernet Address, Protocol, and whether the circuit was created by Automatic Configuration or manually with SYSGEN.

```
Select Option: 1 - Status of Circuit

Cir Status  Link Status Volume Groups Remote System     Mode  Defn
--- ------- ---- ------ ------------- ----------------- ----- ----
VA  Ena,Rea   0  Ena,O  VAX,TST       12-34-56-78-9A-BC DSMV2 Auto
```

Circuit activity display

Use this option to display the activity on a specified circuit. The system displays the number of requests transmitted, requests received, network packets transmitted and received, packets that were retransmitted because of errors, and packets received out-of-sequence. A high number of retries or sequence errors may indicate network problems.

```
Select Option: 2 - Circuit Activity Display

Select Circuit: ?

    Enter the number of the Circuit that is to be displayed, or
    Enter '*' to display all Circuits that are defined, or
    Enter '^L' for a list of the defined Circuits.

Select Circuit: MDC

Ckt   Requests  Requests  Packets   Packets     Xmit  Out-of-Seq
Name   Xmitted  Received  Xmitted  Received  Retries     Packets
----  --------  --------  -------  --------  -------  ----------
MDC      13471     26108    39602     39602        0           0

Select Circuit: RETURN

Select Option:
```

Enable circuit

Use this option to enable a specified circuit in the network. Enter the name of the circuit to be enabled. If the circuit is already enabled, the system responds with an appropriate message and no action takes place.

```
Select Option: Enable Circuit

Select Circuit: ?

    Enter the name of the Circuit that is to be enabled, or
    Enter '*' to enable all Circuits that are defined, or
    Enter '^L' for a list of the defined Circuit.

Select Circuit: VAX

VAX ... Circuit Enabled
```

Disable circuit

Use this option to disable a specified circuit in the network. Enter the name of the circuit to be disabled. If the circuit is already disabled, the system responds with an appropriate message and no action takes place.

```
Select Option: Disable Circuit

Select Circuit: ?

    Enter the name of the Circuit that is to be disabled, or
    Enter '*' to disable all Circuits that are defined, or
    Enter '^L' for a list of the defined Circuit.

Select Circuit: VAX

VAX ... Circuit Disabled

Select Circuit: RETURN

Select Option:
```

Reset statistical counters

Use this option to reset the statistical counters for a specified circuit. Enter the name of the circuit to be reset.

```
Select Option: 5 - Reset Statistical Counters

Select Circuit: ?

    Enter the name of the Circuit that is to be reset, or
    Enter '*' to reset all Circuits that are defined, or
    Enter '^L' for a list of the defined Circuits.

Select Circuit: VAX

VAX ... Circuit Counters Reset

Select Circuit: RETURN

Select Option:
```

Save circuit information

Use this option to save circuit information obtained either from Automatic Network Configuration or from updates to nodes in Manual Configuration mode that have occurred as a result of broadcast messages over the network. Enter the name of the circuit to be saved.

```
Select Option: 6 - Save Circuit Information

Select Circuit: ?

    Enter the name of the Circuit that is to be saved, or
    Enter '*' to save all Circuits that are defined, or
    Enter '^L' for a list of the defined Circuit.

Select Circuit: VAX

VAX ... Circuit Entry Saved

Select Circuit: RETURN

Select Option:
```

This option is useful for setting up complex configurations that will be operated in manual configuration mode. Initially, the system can be run in automatic mode to establish the circuit information. When all desired nodes are connected to the network, the circuit information can be saved. This way, you need not manually enter all of the information into the system.

Communicating with DSM Systems

The MSM system can communicate through the network not only to other MSM systems, but also to DSM (Digital Standard MUMPS). Only Ethernet links can support MSM to DSM communications.

The following sections describe how to set up the DSM and MSM systems to enable communications. This information describes how global variables created on early versions of DEC M software are accessed through the network.

Configuring the DSM System

No special considerations are required when configuring a DSM system to communicate with an MSM system. Follow the same setup procedure that is used to define communications between two DSM systems. To the DSM system, the MSM system behaves like any other DSM system on the network.

When the MSM system is set up, by default all Links are assumed to operate in Automatic Network Configuration mode. In this mode, the MSM system dynamically determines the system type (MSM or DSM) and the release level of the software for each node connected to the network. For all network accesses, the MSM system uses the appropriate communications protocol.

If the system is defined to use Manual Network Configuration mode, you must explicitly define every defined link. After this is done, the system functions as if the Automatic Network Configuration feature was used.

Accessing 7-Bit Globals

When a global node is accessed on a node in the network that is running DSM, the system assumes the global is stored in 8-bit format. If the global variable was created under an old version of DEC M in 7-bit mode, the system cannot access the data. To circumvent this problem, use the Translation feature of MSM to access the global.

When the global is defined in the MSM Translation table, specify 7-bit mode for the global. During the actual global reference, MSM can determine the format of the global using the contents of the Translation table entry. Refer to “Generating the System” in this manual for additional information on this feature.

Application Considerations

Generally, MSM application programs are not aware that data accesses are taking place using DDP or RVG instead of local databases. Because of the buffering that may be performed to ensure efficient processing of DDP and RVG requests, it may be desirable in some situations for the application job to consider the environment of remote databases. These considerations will not have any adverse effect if applied to local databases.

Asynchronous Errors

In the older DDP protocol, MSMV2, each global request was completed and acknowledged by the remote server before the client's M command completed. This ensured that an error occurring in the remote database could be presented to the client job immediately (the error is synchronous with the M command making the database access).

The current DDP protocol, MSMV3, is capable of buffering several global SET and KILL requests into one network packet. Remote Volume Group accesses also may use this buffering technique. When the requests are buffered, the client job does not wait for a response from the remote server. This allows increased network efficiency because of fewer packets and significantly increases the performance of the client job.

The remote server can detect an error at a point after the client believes the request was completed. Such an error is called an *asynchronous error*. Conditions that can lead to asynchronous errors include hardware failure (DKHER), database corruption (DKSER), and database full (DKFUL).

When an asynchronous error is received by the client job, the $ZERROR special variable is set to <ASYNC>location:::6:22:code. Location is the line currently being executed when the error is received (which may be completely unrelated to the source of the error). Code is a number formed from the major and minor error numbers of the error condition that was detected by the remote server. Code is calculated from (major * 256) + minor.

A mode flag is provided that causes the error information to be saved in the client job's pvector instead of interrupting the job with an <ASYNC> error. This mode flag can be set interactively with the %MODESET utility or by the application by calling an entry point of the %DBSYNC utility. The error information can be queried at an appropriate time in the application by using the %DBSYNC utility. Refer to the *MSM Utility Manual* for detailed information on %DBSYNC and %MODESET.

Another method of avoiding asynchronous errors is to wait for the response from the remote system instead of buffering requests. Use the %MODESET utility to select buffering or waiting. Two flags are used. One flag controls processing for DDP requests (MSMV3 protocol). The default mode for DDP requests is to wait for the response. The other flag controls processing for Remote Volume Group requests. The default mode for RVG requests is to buffer the requests and not wait for the response.

ZSYNC Command

The ZSYNC command provides the ability to ensure that all buffered requested have been processed by the remote server. The ZSYNC command provides this function by synchronizing the client job with all remote database requests issued by the job that are still outstanding. Refer to the *MSM Reference Guide* for the syntax of the ZSYNC command.

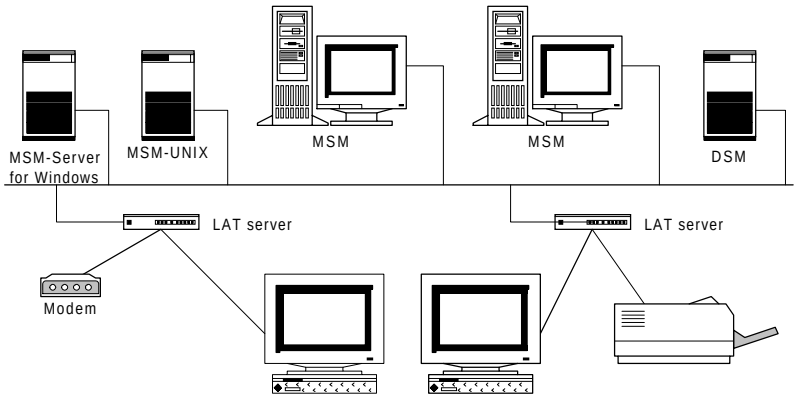
When the ZSYNC command completes, all responses to previously issued requests have been received (either with or without errors), and there is no possibility of an asynchronous error occurring unless a new request is issued. If an error is detected during the ZSYNC processing, an <ASYNC> error occurs immediately if asynchronous errors are enabled. Otherwise, the job's error counter is incremented for later checking by the %DBSYNC utility.

MSM-LAT Terminal Services

Overview

MSM-LAT is an optional package for MSM-UNIX which provides support for the Local Area Transport (LAT) protocol in MSM. MSM-LAT allows terminal, printer, and modem devices to be connected to MSM-UNIX using externally attached terminal servers. The servers, which are linked to the system through an Ethernet (IEEE 802.3) local area network, communicate using the LAT protocol originally developed by Digital Equipment Corporation (DEC). The following figure shows a typical configuration with multiple machines, terminals, printers, and LAT terminal servers and shows how multiple M implementations can be connected to the same terminal network.

LAT Terminal Server Configuration



The LAT protocol, which is available from and supported by numerous manufacturers, provides a flexible and efficient method of attaching terminals to multiple systems through an Ethernet network. The network that is used may be dedicated to LAT or it may be shared by LAT and other communication protocols (for example: IPX or TCP/IP).

A LAT server and MSM communicate by exchanging *packets* (messages) across an Ethernet link. The message format is defined by the LAT protocol licensed by DEC. MSM systems interpret messages that are received and create responses in the proper format to read data from and write data to devices attached to the network.

All message formatting and communication with the terminals is performed in a manner that is totally transparent to the M application. No changes to application programs or to READ and WRITE commands are required to use LAT. Device numbers are used for LAT devices in a manner similar to that used for other MSM devices.

The $ZDEVICE function returns the actual LAT device name for a particular $IO number. This function is intended for applications that are required to know the actual LAT device name. Refer to "MSM Functions" in the *MSM Language Reference Manual* for additional information on the $ZDEVICE function.

Traditionally in a single-system environment, terminals, printers, and modems are directly connected (hard-wired) to the machine through serial I/O cards or communications port cards, multiplexors, and so on. When multiple systems exist within a configuration, terminal wiring must be physically moved or an external switching device must be used. With LAT terminal servers, switching between machines is done through software in the server. This approach is considerably more efficient and effective than manual operations.

Using LAT technology, a simple but elegant solution to terminal networks can be implemented. Because LAT terminal servers can be located long distances from the CPUs they access, terminal wiring distances can be reduced. Other benefits such as redundancy, user-selected switching, or relocation of applications between machines in a transparent manner can be realized. The following sections describe the LAT distribution package, prerequisite system software, the system level software, configuration and management of LAT terminal networks, and the LAT System Management utility programs.

Installing MSM-LAT

MSM-LAT is an optional add-on feature for MSM-UNIX. It includes utility programs and system software which implement the Local Area Transport (LAT) protocol on top of the MSM-NET package. The MSM-LAT utility programs are included with the standard distribution package for MSM-UNIX.

When MSM-LAT is purchased, a new paper key that activates MSM-LAT is provided.

Configuring the LAT Software

Use the SYSGEN utility to define, configure, and maintain all aspects of LAT in MSM. After you invoke SYSGEN, select an existing configuration or create a new configuration.

On the SYSGEN menu for the configuration, select the Network Configuration option. The system responds with the following sub-menu.

```
Select Option: 15 - DDP and LAT Configuration

Available Functions:

     1 - DDP System Parameters
     2 - Link Definition
     3 - Circuit Definition
     4 - DDP Groups
     5 - Network Security
     6 - LAT System Parameters
     7 - LAT Node Management
     8 - LAT HFS Access Control
     9 - OMI Translation Table
    10 - Display Network Configuration
    11 - Workstation Server Configuration
    12 - User-Defined Services
    13 - Telnet Service Configuration
    14 - API Service Configuration
    15 - PDQweb Service Configuration

Select Option:
```

The following sections describe the three LAT options which appear in the menu above. A sample terminal session is provided for each option.

LAT System Parameters

Use this option to specify general system parameters associated with the LAT functions and to tune the LAT system for optimum performance across the network by specifying the number of servers, the maximum number of terminals that can be attached through LAT, and the advertising period for LAT services.

LAT requires additional DDP buffers beyond the amount required for normal DDP operations. Increase the number of DDP buffers by three for each LAT server that will be connected to the system. If the number of DDP buffers is too small, system performance will be adversely affected. Refer to "MSM-NET Services" in this manual for additional information on configuring DDP buffers.

```
Select Option: 6 - LAT System Parameters

Maximum Number Of LAT Servers <3>: ?

    Enter the maximum number of LAT servers that will communicate with
    this system.  A LAT circuit block will be allocated for each
    server.

Maximum Number Of LAT Servers <3>: 8

Maximum Number Of LAT Terminals <30>: ?

    Enter the maximum number of LAT terminals that can be active on
    this system.  Note that control blocks and buffers will be
    allocated for each terminal.

Maximum Number Of LAT Terminals <30>: 64

LAT Advertisement Period In Seconds <60>: ?

    Enter the time, in seconds, for how often this system will
    advertise its LAT services.

LAT Advertisement Period In Seconds <60>: 30

Select Option:
```

When the LAT software is started, the system automatically initiates execution of a LAT advertising job (the DDPADV routine) as a background process. This job periodically sends advertising messages to all servers on the network to indicate which services are available on the system (node). Users connect to a system via connection to an advertised service.

The system manager specifies the frequency for sending advertising messages. Refer to "Services Management" in this chapter for information on services and advertising.

LAT Node Management

Use this option to define LAT groups within MSM, create or edit the LAT node name, and create or edit a description of the LAT node. When you select this option, the system displays a sub-menu of two options.

Use Option 1, LAT Groups, to manage LAT groups within MSM. LAT groups are used to control which servers can attach to specific systems. After a LAT server is defined as belonging to a group, it can only attach to systems that are members of that group.

When LAT groups are used, they must be defined in MSM and in the LAT server. For the connection to be established, the group definitions in MSM and in the server must match. Refer to the appropriate LAT documentation supplied with the LAT server for information on establishing LAT groups within the server.

Use Option 2, LAT Node Management, to define the name of the LAT node. The name, which must be unique, may be up to sixteen characters in length and may include alphabetic and numeric characters. You may enter a description of the node, which is used for informational purposes and to identify the node within the network.

```
Select Option: 7 - LAT Node Management

LAT Node Management Options:

    1 - LAT Groups
    2 - Edit Node Name and Description

Select Option:
```

LAT groups

In MSM-LAT, systems and servers are organized into related groups. A server and a system in the same group may communicate with each other. If a server and a system are not members of the same group, they cannot communicate with each other.

A server or a system may belong to more than one group. In addition to defining groups within MSM, groups must also be defined within each server. For information on defining groups within a server, refer to the appropriate documentation supplied with the LAT terminal server.

```
Select Option: 1 - LAT Groups

Enter LAT Group Number: ?

    Enter a list of LAT group numbers from 0 to 255 to which this
    system will be added as a member. Each system can belong to more
    than one LAT group. LAT servers will only be able to connect to
    systems which share membership in a common LAT group.

    If no groups are specified, then this system automatically is
    assigned membership in LAT group 0.

    LAT servers themselves must also be assigned to LAT groups if this
    feature is to be used. Refer to the documentation for your LAT
    server for instructions on how to assign LAT groups to your server.

    Enter a list of group numbers, separated by commas, to add
    membership to the groups.

    Enter a '-' in front of the list to remove membership from the
    listed groups.
```

```
    Each part of the list may be a range in the form number1-
    number2.

    Each ^L to list the groups to which this system currently
    belongs.

    Enter '^' or RETURN to return to the menu.

Enter LAT Group Number: ^L

    No groups specified, default membership is group 0.

Enter LAT Group Number: 20  ...membership to group 20 added.

Enter LAT Group Number: RETURN

Select Option:
```

Edit node and description

Every system in the LAT network must have a unique name, which may be from one to sixteen characters in length. You can enter a description for the node, which will be displayed by the server in response to the SHOW NODES command.

```
Select Option: 2 - Edit Node Name and Description

Enter Node Name <MSMSML> ?

    This is the name by which this system will be known in the network.
    It must be unique within the LAT network. The maximum length is 16
    characters.

Enter Node Name <MSMSML> RETURN

Enter Node Description <MSM-UNIX, V 4.4.0> ?

    Enter a short description of this node. This will be displayed by
    the server in response to the 'SHOW NODES' command.

Enter Node Description <MSM-UNIX, V 4.4.0> RETURN

Select Option:
```

LAT HFS Access Control

Normally, in order to access MSM, you must first log on to the UNIX system by supplying a valid UNIX userid and optionally, a valid password. Once you are logged on to UNIX, you are assigned file permissions associated with the processes you created. However, if you log on to MSM through LAT, the entire UNIX login process is bypassed. You are not known to UNIX, and UNIX has no file security information about you.

MSM allows you to assign certain UNIX user attributes to MSM-LAT users, so that access to UNIX files via the Host File Server (HFS) interface of MSM can be controlled. These attributes include User Name (for new and existing files), File Permissions (for new files), and File Size limit (for new files).

```
Select Option: 8 - LAT HFS Access Control

Enter User Name: ?

    Enter the UNIX user name for LAT users. When accessing UNIX files,
    LAT users will be treated as if they had this user name for file
    access purposes.

Enter User Name: guest

Enter File Permissions <rw-rw-rw->: ?

    Enter the UNIX permissions to be assigned to any file created by a
    LAT user.

    Access to UNIX files for read, write, and execution is allowed
    depending on the user and group id of the process attempting the
    access.  The file permissions are set using a symbolic mask of nine
    (9) characters that correspond to user classes and the permissions
    for each class.  The first three characters control user
    permissions (i.e., the owner of the file), the next three control
    group permissions, and the last three control permissions for users
    who do not  belong the group that owns the file.

    Within each group of three letters, only one of two characters are
    allowed at each position.  The first position must be an "r",
    indicating read access is allowed, or a "-" (i.e., a hyphen),
    indicating read access is not allowed.

    The second position must be either a "w" for write access, or a
    hyphen. The third position must be either an "x" for execute
    access, or a hyphen.  Consider the following example.

              rwxr-x---

    This mask indicates that the owner of the file can read, write, and
    execute the file, while others in the same group can read and
    execute the file, but not update it.  All other users cannot access
    the file at all.

Enter File Permissions <rw-rw-rw->: rw-rw----

Enter File Size Limit <8192K>: ?

    Enter the file size limit for LAT users.
    Enter nnnnK to assign the value in K bytes.
    Enter nnnnM to assign the value in megabytes.
    This feature is the equivalent of the ulimit parameter for UNIX
    users.

Enter File Size Limit <8192K>: RETURN

Select Option:
```

LAT Servers and Devices

This section describes the naming conventions for servers and for ports attached to servers, describes the system logon procedure, provides an overview of the Host-Initiated Connection feature, and provides guidelines for configuring terminal servers. It is assumed that you are familiar with the capabilities of LAT terminal servers.

Server and Port Names

To facilitate LAT terminal network management, each terminal server in the network and each terminal port that is attached to the servers must have unique identification. In order to ensure uniqueness, LAT terminal servers are assigned a name in the factory that includes the Ethernet address of the device. For example:

LAT_0000B5000A98

The server name is formed by concatenating the word LAT with an underscore (_) and the Ethernet address of the terminal server. The hyphens are removed from the Ethernet address before the concatenation takes place. Similarly, ports are assigned names like the following:

PORT_2

The port name is formed by concatenating the word PORT with the actual number of the port on the terminal server. Within MSM, the unique device name for a port attached to a terminal server is in the following format:

PORT_2@LAT_0000B5000A98

Th device name is formed by concatenating the port name with an at-sign (@) and the LAT terminal server name. Because these device names can be somewhat awkward, terminal servers allow you to edit the server and port names to make them more meaningful. For example, the above device name could be edited to appear like one of the following:

PRINTER3@LABSERVER

PORT_2@SERVER_03

In MSM, use the SYSGEN utility to assign LAT server device names to standard I/O device numbers. After this is done, application programs can OPEN, USE, and CLOSE ports by their device number. For applications that need to know the actual device name, use the $ZDEVICE function. For information on the $ZDEVICE function, refer to “MSM Functions” in the *MSM Language Reference Manual*. For information on the SYSGEN utility, refer to “Generating the System” in this manual.

Logging on to the System

When you log on to the system through a LAT device, you are first connected directly to the terminal server. At the server’s command prompt, enter any valid server command.

In an application environment, the advertised services could be tied to a particular routine on a specific Volume Group. “Services Management” in this chapter describes how terminals can be tied to an application.

Printers and Host-Initiated Connects

MSM supports the *Host-Initiated Connection (HIC)* feature which allows you to share devices (such as impact printers, laser printers, or outbound modems) between multiple systems. This feature allows applications to OPEN a device that is connected to a LAT terminal server.

A Host-Initiated Connection is initiated from the host computer to the terminal server. Connections initiated from the terminal server to the host computer are referred to as *User-Initiated Connections.* In MSM, you can set up ports to allow Host-Initiated Connections, User-Initiated Connections, or both.

For User-Initiated Connects, you need not define the device in SYSGEN. For Host-Initiated Connects , you must define the port in SYSGEN. Define each port as a LAT-type port and enter the LAT server device name (for example: PORT_2@SERVER_03). Additional information on defining LAT devices in SYSGEN is provided in "Generating the System" in this manual.

Devices that are used for Host-Initiated Connections must be set up properly in the LAT server. "Recommended Service Parameters" in this chapter provides suggested settings for various classes of devices. Error conditions may occur even if the device is configured properly. The following list of common error conditions explains why such errors occur:

1. *<NODEV> error*: Usually indicates that the device is not defined in SYSGEN. It also can occur if the maximum number of LAT terminal connects is exceeded.
2. *OPEN command with a timeout expires or an OPEN without timeout never completes*: Occurs if the LAT server device name is misspelled in SYSGEN or if the port characteristics in the server are incorrect (for example: access is LOCAL rather than REMOTE or DYNAMIC). It also can occur if the terminal attached to the server is at the server command prompt (for example: >Local prompt) or if the device is currently OPENed by another job.
3. *Unreadable Output*: Generally caused by a mismatch between the terminal server port characteristics and the device characteristics (for example: baud rate or parity).

Recommended Server Parameters

The parameter settings in the LAT Server must be set properly for the intended use of the device that will be attached to the port. If the settings are not correct, error conditions may occur within MSM-LAT (such as spurious terminal disconnects or inability to log on to the system).

The following table provides general guidelines for setting LAT server parameters to accommodate various classes of devices. The parameters in this table are not defined because it is assumed that you are familiar with them.

Recommended LAT Server Parameters

ServerParameters	Terminals[1]	Modems[2]	Printers3
Access	Local	Dynamic	Remote
Enabled	Autobaud	Disabled	Disabled
Autoprompt	Enabled	Disabled	Disabled
Autoconnect	Disabled	Disabled	Disabled
Preferred	None	None	None
Dedicated	None	None	None

Notes:

1. Terminals can be used to log on to the system. They can be CRT-type terminals or hard copy-type terminals. Host-Initiated Connections are not allowed.
2. Modems can be used to log on to the system or to allow Host-Initiated Connections to dial-out from the system.
3. Printers are considered to be output-only devices and are only available for Host-Initiated Connections.

In addition to the parameters shown in the table above, other parameters must be set, including Baud Rate, Parity, and Stop Bits. Refer to the documentation supplied with the terminal server for a description of these and other server parameters.

The LAT Utility

The LAT feature of MSM includes a system management utility (the LAT utility) that is used to control a node's access to the network (startup and shutdown), define and manage LAT services, and perform maintenance functions at the Circuit level.

Invoking the LAT Utility

To access the LAT utility, you must be logged on to the Manager's UCI in programmer mode. When the LAT utility is invoked, the system displays the menu shown in the following example.

```
>D ^LAT

          MSM - Local Area Transport Utility

Available Functions:

    1 - Start up LAT and DDP
    2 - Shut down LAT and DDP
    3 - Services Management
    4 - Link Management
    5 - LAT Circuit Management

Select Option: ?

    Select option by specifying the option number or supplying enough
    characters to uniquely identify the option,
    To get help information specific to an option, enter a '?' followed
    by the option's number,
    Enter <RETURN> or '^' to exit with no action,
    Enter '^L' to get a list of options

Press <RETURN> to continue

Select Option:
```

Start Up LAT

Use this option to start LAT processing on the machine. When this option is invoked, the system automatically starts the DDP networking portion of LAT. Alternatively, use the DDP utility to start DDP; LAT is automatically started when DDP is started. Refer to "MSM-NET Services" in this manual for additional information on the DDP utility and starting links.

```
Select Option: 1 - Start up LAT

How Many Servers Do You Want To Invoke <1>: ?

    Enter the number of DDP servers to be started. Busy systems may
    require more than 1 server in order to maintain optimum throughput
    on the network.

How Many Servers Do You Want To Invoke <1>: RETURN

DDP Startup ...  Link 1 is now enabled

Select Option:
```

Shut Down LAT

Use this option to stop LAT. When you select this option, all active Links on the machine are stopped. The shutdown process broadcasts a message to all other nodes on the network as each Link is stopped. Terminals connected through LAT will be disconnected.

```
Select Option: 2 - Shut down LAT

OK to stop DDP and LAT <N>? Y

DDP Shutdown ...  Link 1 is now disabled

Select Option:
```

Services Management

In MSM, a LAT network provides services to users. A *service* usually refers to a specific system function (such as an application program) or a particular system. A unique service name is associated with every LAT service. The LAT protocol allows individual servers to create and manage logical connections between ports on the server and defined services. These logical connections are referred to as *sessions*.

Use this option to create, edit, and delete LAT services and to display defined LAT for the system. When this option is selected, the following menu is displayed.

```
Select Option: 3 - Services Management

Available Functions:

    1 - Edit Services List
    2 - Display Services List
    3 - Enable Volume Group Services Advertising
    4 - Disable Volume Group Services Advertising

Select Option:
```

LAT services are associated with a particular volume group. The service definitions are stored in the ^SYS global of UCI number 1 on the volume group. Therefore, no LAT services may be defined on a volume group until the first UCI is defined.

When the LAT system parameters are initially set, LAT service number 1 is automatically created in the system volume group. The name of this service is MSM-XXX, where XXX is the name of the system volume group. This service is reserved for programmer mode access (no login string may be specified), and it may not be deleted. If programmer mode access is not needed, the service may be disabled.

When LAT is started, all enabled services on all mounted volume groups are advertised. If a volume group is mounted later that contains enabled LAT services, then those services are added to the advertisement. When services are edited, any changes are advertised immediately.

Edit services list

Use this option to create, edit, and delete services. A service consists of a service name, a login string to be used when the service is invoked, and a description.

```
Select Option: 1 - Edit Services List

Service #   Name      Status    Login String   Description
---------   -------   -------   ------------   -----------------
1           MSM-SML   Enabled                  MSM-UNIX, V 4.4.0
2           LAB       Enabled   LAB:MENU       LABORATORY SYSTEM

Enter Service Number: 2

Enter Service Name <LAB> ?

    Every LAT Service has a service name.  The user must use this name
    to connect to this system. The maximum length of the name is 14
    characters.

Enter Service Name <LAB> RETURN

Enabled (Y or N) ? <Y> ?

    Enter 'Y' to ENABLE the service.
    Enter 'N' to DISABLE the service.
    Only enabled services will be advertised.

Enabled (Y or N) ? <Y> RETURN

Enter Login String <LAB:MENU> ?

    Enter the login string that will automatically be passed to MSM
    when this service is selected.  The format of the string is
    'uci:routine:size'. When a terminal connects to this service, it
    will automatically be connected to the specified UCI and the
    routine will be invoked. The size parameter is optional and is used
    to override the default system partition size. This is equivalent
    to using the tied terminal facility.

    To provide access to programmer mode from this service, enter
    MGR:%LOGON for the login string.

Enter Login String <LAB:MENU> RETURN

Enter Service Description <LABORATORY SYSTEM> ?

    Enter a short description for this service. This will be displayed
    by the server in response to the 'SHOW SERVICES' command.

Enter Service Description <LABORATORY SYSTEM> RETURN

Enter Service Number: RETURN

Select Option:
```

The server displays the service name and description in response to the SHOW SERVICES command. The specified logon string is passed to MSM as if it was entered in response to the logon prompt. Refer to “Using the MSM System” in the *MSM User's Guide* for information on the format of responses to the logon prompt.

Display services list

Use this option to display the list of defined LAT services. For each service, the service number, name, status, login string, and description are displayed. Service number 1 on volume group 0 has no login string; it connects directly to the %LOGON utility.

```
Select Option: 2 - Display Services List

Services in Volume Group:  PRD

Service #   Name      Status    Login String   Description
---------   -------   -------   ------------   -----------------
1           MSM-SML   Enabled                  MSM-UNIX, V 4.4.0
2           LAB       Enabled   LAB:MENU       LABORATORY SYSTEM

Select Option:
```

Enable volume group services advertising

Use this option to enable advertising for LAT services in a selected volume group.

```
Select Option: 3 - Enable Volume Group Services Advertising

Services in Volume Group : ABC     Advertising: Disabled

Service #  Name     Status   Login String    Description
---------  -------- -------- -------------   ----------------------
1          MSM-SML  Disabled                 MSM-Server for UNIX

Select volume group <ABC>: RETURN ..Enabled.
```

Disable volume group services advertising

Use this option to disable advertising for LAT services in a selected volume group.

```
Select Option: 4 - Disable Volume Group Services Advertising

Services in Volume Group : ABC     Advertising: Enabled

Service #  Name     Status   Login String    Description
---------  -------- -------- -------------   ----------------------
1          MSM-SML  Enabled                  MSM-Server for UNIX

Select volume group <ABC>: RETURN

Select volume group <ABC>:    ...Disabled.
```

Link Management

Use this option to control functions associated with the communications link and to display status information about the link. When you select this option, the following menu is displayed.

```
Select Option: 4 - Link Management

Available Options:

    1 - Status of Link
    2 - Enable Link
    3 - Disable Link
    4 - Change Link Status

Select Option:
```

You also can access this function using the DDP utility. Refer to the DDP utility in “MSM-NET Services” in this manual.

LAT Circuit Management

Use this option to control functions associated with individual LAT communications circuits. You may display circuit status and activity, display the statistical counters associated with a circuit, or reset the statistical counters associated with a circuit. When this option is selected, the following menu is displayed.

```
Select Option: 5 - LAT Circuit Management

Available Functions:

    1 - Status of LAT Circuits
    2 - Display LAT Circuit Counters
    3 - Reset LAT Circuit Counters

Select Option:
```

Status of LAT circuits

Use this option to display the status of defined LAT circuits. It includes Circuit Number, Circuit Name, Ethernet Address, and the Circuit status (Active or Inactive).

```
Select Option: 1 - Status of LAT Circuits

Cir. No.  Circuit Name   Ethernet Address     Status
--------  ------------   -----------------    ------
    0     VISTA          00-00-B5-00-0A-98    Active

Select Option:
```

Display LAT circuit counters

Use this option to display the counters for a specified LAT circuit. It displays information on circuit activity, number of messages transmitted, number of messages received, number of messages retransmitted because of errors, and so on. This information is obtained directly from the server. For additional information on the contents of the display, refer to the appropriate server documentation.

```
Select Option: 2 - Display LAT Circuit Counters

Cir No.   Circuit Name   Ethernet Address    Status
-------   ------------   -----------------   ------
   0      VISTA          00-00-B5-00-0A-98   Active

Enter Circuit Number : 0

Server: VISTA        Status: Active             Slots Connected: 1

VC Block: 232BE54    Rem: 00-00-B5-00-0A-98 Loc: 00-00-C0-47-22-18

Retransmit events:       0     Seconds after Reset:      236
Retransmitted messages:           0

                         Received        Sent
Run messages, total:           29          29
Run messages, empty:           11          20
Data_a Slots, total:           16          10
Data_a Slots, empty:            8           0
Data_a characters:              8          67
Out-of-sync:                    0                  pwa:         0
Duplicates:                     0                  iwa:         0
Not all acked:                  0
Illegal messages:               0
Illegal Slots:                  0

Enter Circuit Number : RETURN

Select Option:
```

Reset LAT circuit counters

Use this option to reset the statistical counters maintained for a particular LAT circuit. When you select this option, the system displays a list of circuits. Enter the number of the circuit to be reset.

```
Select Option: 3 - Reset LAT Circuit Counters

Cir No.  Circuit Name   Ethernet Address   Status
-------  ------------   -----------------  ------
   0     VISTA          00-00-B5-00-0A-98  Active

Enter Circuit Number : 0  ... counters reset

Enter Circuit Number : RETURN

Select Option:
```

Supported LAT Servers

The MSM-LAT feature works with all popular LAT terminal servers that allow their software to be permanently loaded into Read-Only Memory (ROM). In addition, MSM-LAT works with all popular PC terminal emulation software that supports the LAT protocol.

Glossary

ANSI

The **A**merican **N**ational **S**tandard **I**nstitute.

argument

An expression which determines the action that will occur in the function or command with which it is used.

array

An organized set of local or global elements or nodes referenced by subscripts and a common variable name.

ASCII

The **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. This code consists of 128 characters which comprise the standardized character set.

background job

A job started by the JOB command. This process runs in parallel with the process that contained the JOB command.

baud

The data transmission rate between two devices.

break

A command used to interrupt program execution. It typically is used to allow debugging to take place.

canonic number

A numeric value which has no leading zeros or trailing zeros after the decimal point.

carriage return

A keyboard instruction, often used to indicate the end of a command. This key commonly is marked RETURN or ENTER.

collating sequence

An order assigned to a group of subscripts. Sorting is done in either string or numeric sequence.

String - All subscripts are treated as character strings and are stored in ASCII sequence.

Numeric - Storage is in the order of canonic numbers first, followed by the non-numeric values in ASCII sequence.

command

The method by which MSM is directed to perform a specific action.

comment

A brief phrase in the body of a routine which describes when the routine was written, what it does, and so on. This non-executable segment of code begins with a semicolon (;) and includes the remainder of the line.

compiler

A highly sophisticated system program which scans each line of M code, divides it into basic components, analyzes each component to ensure that it is syntactically correct, and generates pseudo-machine code that can be processed by the p-code interpreter.

concatenation

The process of joining two operands together to form a new string. This is accomplished by using the underline symbol (_) to append operand2 to the end of operand1.

CONFIG.msm

A file located in the MSM directory which contains configuration information about the MSM system. When MSM is started, this file is read to determine how MSM is to be set up.

configuration

The collection of hardware and software that comprises the entire computer system.

control characters

Characters from the standard ASCII set which have special meaning to the MSM system. To obtain these characters, press the CTRL key while pressing the associated control character.

cursor

The on-screen marker, usually a box or an underline, which indicates the location where the next data entry will occur.

data

Information (letters, numbers, symbols) which is entered into the system for processing or storage.

database

The location where data storage takes place in global arrays. In MSM, this is where the MSM system is installed (database 0).

default

A value which is assumed as the entry to a prompt if RETURN is pressed. In MSM utilities, default values are displayed within greater than and less than signs (for example: <DEFAULT>).

descendant

Any array node on a lower subscript level which can be reached from that node and which shares the first x subscripts in common. For example, the nodes R(3,4,5) and R(3,6,4,7) are descendants of R(3).

device

Any part of the computer other than the CPU, the memory, or any associated architectural part; for example: a printer, terminal, or modem.

expression

A character string which yields a value upon execution.

function

An action which enables you to perform routine operations more efficiently. In MSM, a function begins with a dollar sign ($), two dollar signs ($$), or with a dollar sign and the letter Z ($Z) for specialized M functions.

global

The permanent data storage used by M. Information is stored in a global array or a simple global variable and generally is placed on a disk system.

global variable

A reference name for data stored in a global on the disk. These variables can be accessed or modified by any user on the system with the proper authorization level.

hardware

The physical components of the computer system other than the software, for example: the computer itself (monitor, disk drive, and so on), the tape drive, and the printer.

indirection

A method of using the value of an expression rather than the expression itself. The symbol used to indicate that indirection is being used is the at-sign (@), followed by the value which represents the expression.

I/O supervisor

An MSM facility which is responsible for the input, error detection, error correction, and output operation of each device attached to the system. For each device type, the I/O supervisor coordinates all data transfers and synchronizes them with job execution.

interpreter

A part of the MSM system which processes the pseudo-machine code generated by the compiler.

job

An M partition separately managed by MSM with respect to execution and I/O.

journaling

A method of recording global SETs and KILLs while the system is in use. All information is recorded in a journal entry which can be used to reconstruct the database.

line

A string of characters ending with a specified READ terminator (carriage return/line feed).

line label

An optional name at the beginning of a routine line which identifies the line to the system. This label is limited to eight characters and must begin with an alphabetic or percent (%) character. Optionally, the label may contain parameter passing variables.

literal

A string, enclosed in quotations, which can be acted upon but not changed by a command. Although the literal may contain any valid ASCII character, certain characters may be excluded because they have special meaning to the MSM system.

local variable

Any variable that exists only within memory. Local variables are unique to a particular job and are valid only for the duration of the job's execution.

map

Space allocated for disk storage. Each map consists of 512 blocks (each 1024 bytes in size).

modem

Acronym for MOdulator DEModulator. This device is used to convert data to a form that can be transmitted via a phone line to a remote site and then reconverted to a form usable by the remote site processing system.

MUMPS

Acronym for **M**assachusetts General Hospital **U**tility **M**ulti-**P**rogramming **S**ystem. This system was developed in the late 1960s to handle storage, retrieval, and manipulation of large amounts of medical data.

naked reference

A shortcut method for referring to a specific global node. A naked reference generally can be used wherever a global reference is permitted. It can be specified for a previously referenced node by using a circumflex (^) followed by the unique portion of the descendant's subscript.

node

An element of a global array which is addressed by the name common to all members of the array and a unique subscript.

operating system monitor

A system which provides an interface between the host operating system and MSM. The monitor also ensures that all system resources are properly allocated among users and maintains overall system efficiency and throughput.

programmer access code (PAC)

A three-letter designation which must be entered in order to gain access to programmer mode. One PAC can be specified for each configuration on the system.

parameters

A collection of guidelines for usage of a particular device.

partition

An area of memory consisting of a logical grouping of the local symbol table, current routine edit buffer, and work areas used by the system and the job. This space expands and shrinks based on the current job's requirements.

peripherals

A collection of physical and logical devices which are associated with the MSM system. They can be used to gain access to printers, terminals, and sequential devices, and to examine or modify memory contents or alternate disk storage.

programmer mode

A mode which enables you to directly enter M commands to the interpreter. To gain access to programmer mode, you must enter a valid UCI and PAC.

prompt

A system message which requires user input.

routines

Library Routines - Utility programs which are accessible to all users on the system.

System Manager Routines - Utility programs which are only accessible via the Manager's UCI.

run mode

A method used to directly access a routine in a particular UCI. To access run mode, you must enter a valid UCI and the name of a routine stored in that UCI.

sparse array

An array which contains space only for defined elements.

special variables

A group of variables (for example: $ZA, $ZB, $ZC) which have special meaning to the MSM system. These variables are used to indicate status information about the results of the last operation performed.

stack

An area of the MSM system set aside for nesting of subroutines resulting from execution commands.

stap

An area of the MSM system set aside for expression nesting.

string

Any set of ASCII characters.

string literal

A string of characters enclosed in double quotes within the context of a command line.

subroutine

A collection of commands which together allow control to pass from the routine to the subroutine and back to the main routine. Subroutines generally are used when multiple recurring tasks are required for execution of the main routine.

subscript

A numeric or string interpreted value appended to a local or global variable. A subscript identifies a specific element or node in an array. Subscripts must be enclosed in parentheses. When multiple subscripts are specified, they are separated by commas.

SYSGEN

The system generation utility which is used to specify one or more configurations to the MSM system. These configurations are used by the system at startup to initialize the system with the defined parameters.

terminators

A specified set of control characters used to terminate a READ operation. MSM uses a default value of line feed, carriage return (the $C(10,13) sequence), but allows you to alter this value by using proper parameters associated with the current device.

tied terminal

A mode which forces a terminal to automatically start up a particular routine in a specified UCI.

timeout

A timing convention used to specify how long the system should attempt to perform a given command before continuing. It is expressed in the format of a colon (:), followed by an integer which is appended to a READ, OPEN, LOCK, or JOB command.

user class identifier (UCI)

A designation (consisting of three uppercase letters) for a work area within the MSM system. Each UCI has its own unique routine and global directory.

utilities

library utilities - Utilities that aid in the development of application programs in which commonly performed functions are required.

system manager utilities - Utilities intended for use by the system manager to ensure proper system performance. They can be accessed only from the Manager's UCI.

variable

A symbolic name for a location where data is stored. In MSM, there are three type of variables:

local variables - Variables stored in memory only.

global variables - Variables stored in arrays for permanent storage on disk.

special variables - Variables which hold special meaning to the MSM system.

volume group (VolGroup)

A string value which specifies a volume group name consisting of three uppercase letters.

volume number (VolNum)

A numeric expression which specifies an internal volume group number.

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Z