

# VMS SYSMAN Utility Manual

Order Number: AA-LA26B-TE

**June 1990**

This document describes the VMS System Management (SYSMAN) Utility, a tool that centralizes the management of nodes and clusters.

**Revision/Update Information:** This manual supersedes the *VMS SYSMAN Utility Manual, Version 5.0.*

**Software Version:** VMS Version 5.4

**digital equipment corporation  
maynard, massachusetts**

---

**June 1990**

---

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

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

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

---

© Digital Equipment Corporation 1990.

All Rights Reserved.  
Printed in U.S.A.

---

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

|            |             |                |                  |
|------------|-------------|----------------|------------------|
| CDA        | DEQNA       | MicroVAX       | VAX RMS          |
| DDIF       | Desktop-VMS | PrintServer 40 | VAXserver        |
| DEC        | DIGITAL     | Q-bus          | VAXstation       |
| DECdtm     | GIGI        | ReGIS          | VMS              |
| DECnet     | HSC         | ULTRIX         | VT               |
| DECUS      | LiveLink    | UNIBUS         | XUI              |
| DECwindows | LN03        | VAX            |                  |
| DECwriter  | MASSBUS     | VAXcluster     | <b>digital</b> ™ |

The following is a third-party trademark:

PostScript is a registered trademark of Adobe Systems Incorporated.

ZK4628

---

## Production Note

This book was produced with the VAX DOCUMENT electronic publishing system, a software tool developed and sold by Digital. In this system, writers use an ASCII text editor to create source files containing text and English-like code; this code labels the structural elements of the document, such as chapters, paragraphs, and tables. The VAX DOCUMENT software, which runs on the VMS operating system, interprets the code to format the text, generate a table of contents and index, and paginate the entire document. Writers can print the document on the terminal or line printer, or they can use Digital-supported devices, such as the LN03 laser printer and PostScript printers (PrintServer 40 or LN03R ScriptPrinter), to produce a typeset-quality copy containing integrated graphics.



---

# Contents

---

|  |                |           |
|--|----------------|-----------|
|  | <b>PREFACE</b> | <b>ix</b> |
|--|----------------|-----------|

---

|  |                           |             |
|--|---------------------------|-------------|
|  | <b>SYSMAN Description</b> | <b>SM-1</b> |
|--|---------------------------|-------------|

---

|          |  |             |
|----------|--|-------------|
| <b>1</b> | <b>DEFINING A MANAGEMENT ENVIRONMENT</b>       | <b>SM-2</b> |
| 1.1      | Defining Another Node as the Environment _____ | SM-3        |
| 1.2      | Defining a Cluster Environment _____           | SM-4        |

---

|          |                                       |             |
|----------|---------------------------------------|-------------|
| <b>2</b> | <b>CENTRALIZING SYSTEM MANAGEMENT</b> | <b>SM-4</b> |
|----------|---------------------------------------|-------------|

---

|          |  |             |
|----------|--|-------------|
| <b>3</b> | <b>ADJUSTING YOUR PROFILE</b>                    | <b>SM-5</b> |
| 3.1      | Modifying Your Current Privileges _____          | SM-6        |
| 3.2      | Resetting the Default Device and Directory _____ | SM-6        |
| 3.3      | Setting DCL Verification _____                   | SM-7        |

---

|          |  |             |
|----------|--|-------------|
| <b>4</b> | <b>EXECUTING COMMANDS</b>                            | <b>SM-7</b> |
| 4.1      | DCL Commands _____                                   | SM-8        |
| 4.2      | SYSMAN Command Procedures _____                      | SM-8        |
| 4.3      | DISKQUOTA Commands _____                             | SM-8        |
| 4.3.1    | Contents of a Quota File • SM-9                      |             |
| 4.3.2    | Establishing Disk Quotas • SM-9                      |             |
| 4.3.3    | Quota Enforcement • SM-10                            |             |
| 4.4      | Cluster Parameter Commands _____                     | SM-10       |
| 4.5      | System Parameter Commands _____                      | SM-10       |
| 4.6      | Startup Commands _____                               | SM-11       |
| 4.7      | Loadable Image Commands _____                        | SM-12       |
| 4.8      | LICENSE Commands _____                               | SM-12       |
| 4.9      | Auto-Login Facility (ALF) Commands _____             | SM-13       |
| 4.10     | SPAWN and ATTACH Commands _____                      | SM-13       |
| 4.11     | Defining Keys to Execute SYSMAN Commands _____       | SM-14       |
| 4.12     | Executing Commands from an Initialization File _____ | SM-14       |
| 4.13     | Executing Commands on Remote Nodes _____             | SM-14       |

---

|          |                        |              |
|----------|------------------------|--------------|
| <b>5</b> | <b>COMMAND SUMMARY</b> | <b>SM-15</b> |
|----------|------------------------|--------------|

---

|  |                             |              |
|--|-----------------------------|--------------|
|  | <b>SYSMAN Usage Summary</b> | <b>SM-18</b> |
|--|-----------------------------|--------------|

---

**SYSMAN Commands**

**SM-20**

|                                  |              |
|----------------------------------|--------------|
| <b>@ (EXECUTE PROCEDURE)</b>     | <b>SM-21</b> |
| <b>ALF ADD</b>                   | <b>SM-23</b> |
| <b>ALF REMOVE</b>                | <b>SM-25</b> |
| <b>ALF SHOW</b>                  | <b>SM-26</b> |
| <b>ATTACH</b>                    | <b>SM-27</b> |
| <b>CONFIGURATION SET</b>         |              |
| <b>CLUSTER_AUTHORIZATION</b>     | <b>SM-29</b> |
| <b>CONFIGURATION SET TIME</b>    | <b>SM-31</b> |
| <b>CONFIGURATION SHOW</b>        |              |
| <b>CLUSTER_AUTHORIZATION</b>     | <b>SM-33</b> |
| <b>CONFIGURATION SHOW TIME</b>   | <b>SM-34</b> |
| <b>DEFINE</b>                    | <b>SM-35</b> |
| <b>DISKQUOTA ADD</b>             | <b>SM-38</b> |
| <b>DISKQUOTA CREATE</b>          | <b>SM-40</b> |
| <b>DISKQUOTA DELETE</b>          | <b>SM-42</b> |
| <b>DISKQUOTA DISABLE</b>         | <b>SM-43</b> |
| <b>DISKQUOTA ENABLE</b>          | <b>SM-44</b> |
| <b>DISKQUOTA MODIFY</b>          | <b>SM-45</b> |
| <b>DISKQUOTA REBUILD</b>         | <b>SM-47</b> |
| <b>DISKQUOTA REMOVE</b>          | <b>SM-48</b> |
| <b>DISKQUOTA SHOW</b>            | <b>SM-50</b> |
| <b>DO</b>                        | <b>SM-52</b> |
| <b>EXIT</b>                      | <b>SM-54</b> |
| <b>HELP</b>                      | <b>SM-55</b> |
| <b>LICENSE LOAD</b>              | <b>SM-56</b> |
| <b>LICENSE UNLOAD</b>            | <b>SM-57</b> |
| <b>PARAMETERS DISABLE CHECKS</b> | <b>SM-58</b> |
| <b>PARAMETERS ENABLE CHECKS</b>  | <b>SM-59</b> |
| <b>PARAMETERS SET</b>            | <b>SM-60</b> |
| <b>PARAMETERS SHOW</b>           | <b>SM-62</b> |
| <b>PARAMETERS USE</b>            | <b>SM-66</b> |
| <b>PARAMETERS WRITE</b>          | <b>SM-67</b> |
| <b>SET ENVIRONMENT</b>           | <b>SM-68</b> |
| <b>SET PROFILE</b>               | <b>SM-70</b> |
| <b>SET TIMEOUT</b>               | <b>SM-71</b> |
| <b>SHOW ENVIRONMENT</b>          | <b>SM-72</b> |
| <b>SHOW KEY</b>                  | <b>SM-73</b> |
| <b>SHOW PROFILE</b>              | <b>SM-74</b> |
| <b>SHOW TIMEOUT</b>              | <b>SM-75</b> |
| <b>SPAWN</b>                     | <b>SM-76</b> |
| <b>STARTUP ADD</b>               | <b>SM-80</b> |
| <b>STARTUP DISABLE</b>           | <b>SM-82</b> |
| <b>STARTUP ENABLE</b>            | <b>SM-84</b> |
| <b>STARTUP MODIFY</b>            | <b>SM-86</b> |

|                      |       |
|----------------------|-------|
| STARTUP REMOVE       | SM-88 |
| STARTUP SET DATABASE | SM-89 |
| STARTUP SHOW         | SM-90 |
| SYS_LOADABLE ADD     | SM-92 |
| SYS_LOADABLE REMOVE  | SM-94 |

---

**INDEX**

---

**FIGURES**

|      |                                   |      |
|------|-----------------------------------|------|
| SM-1 | Defining a Management Environment | SM-2 |
|------|-----------------------------------|------|

---

**TABLES**

|      |                     |       |
|------|---------------------|-------|
| SM-1 | Key Names in SYSMAN | SM-35 |
|------|---------------------|-------|





---

## Preface

---

### Intended Audience

This document is intended for VMS system managers—especially those who maintain two or more VMS nodes. The manual shows how to define a system management environment, modify your process attributes for the environment, and perform management tasks. To use SYSMAN functions you should know how to use DCL commands and be familiar with VAXcluster management requirements.

---

### Document Structure

This document consists of the following three sections:

- Description—Provides a full description of the SYSMAN Utility.
- Usage Summary—Outlines the following SYSMAN information:
  - Invoking the utility
  - Exiting from the utility
  - Restrictions or privileges required
- Commands—Describes SYSMAN commands, including format, parameters, and examples.

---

### Associated Documents

For descriptions of the DCL system management commands, see the *VMS DCL Dictionary*.

For an overview of VMS system management tasks, see the *Introduction to VMS System Management*. The *Guide to Setting Up a VMS System* and the *Guide to Maintaining a VMS System* contain descriptions of day-to-day VMS system maintenance tasks.

For additional cluster information, see the *VMS VAXcluster Manual* and the *VMS Show Cluster Utility Manual*.

---

### Conventions

This manual uses the following conventions:

Ctrl/x

A sequence such as Ctrl/x indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.

Return

In examples, a key name is shown enclosed in a box to indicate that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)

## Preface

|                |   |
|----------------|---|
| ...            | In examples, a horizontal ellipsis indicates one of the following possibilities: <ul style="list-style-type: none"><li>• Additional optional arguments in a statement have been omitted.</li><li>• The preceding item or items can be repeated one or more times.</li><li>• Additional parameters, values, or other information can be entered.</li></ul> |
| .              | A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.   |
| ()             | In format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.   |
| []             | In format descriptions, brackets indicate that whatever is enclosed is optional; you can select none, one, or all of the choices. (Brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)   |
| { }            | In format descriptions, braces surround a required choice of options; you must choose one of the options listed.  |
| red ink        | Red ink indicates information that you must enter from the keyboard or a screen object that you must choose or click on.<br><br>For online versions of the book, user input is shown in <b>bold</b> .   |
| UPPERCASE TEXT | Uppercase letters indicate that you must enter a command (for example, enter OPEN/READ), or they indicate the name of a routine, the name of a file, the name of a file protection code, or the abbreviation for a system privilege.  |
| -              | Hyphens in coding examples indicate that additional arguments to the request are provided on the line that follows.   |
| numbers        | Unless otherwise noted, all numbers in the text are assumed to be decimal. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.   |

---

## SYSMAN Description

The System Management Utility (SYSMAN) centralizes the management of nodes and VAXcluster environments. Rather than logging in to individual nodes and repeating a set of management tasks, SYSMAN allows you to define your management environment to be a particular node, a group of nodes, or a VAXcluster environment. With a management environment defined, you can perform system management tasks from your local node. SYSMAN executes these tasks on all nodes in the target environment.

Managing a system with SYSMAN is similar to the traditional management of an individual system because SYSMAN uses many of the same software tools. It can process most DCL commands, such as MOUNT, DEFINE, INITIALIZE, SET, and SHOW. It can also execute many VMS system management utilities and command procedures, such as AUTHORIZE, AUTOGEN, and INSTALL.

SYSMAN also contains system management tools that let you perform the following tasks:

- Set disk quotas using DISKQUOTA commands.
- Load and unload licenses using LICENSE commands.
- Associate a terminal or port with a user name using the Auto-Login Facility (ALF) commands.
- Modify or display system (SYSGEN) parameters using PARAMETERS commands.
- Build site-specific startup procedures using STARTUP commands, which display or modify startup components of the VMS operating system, site-specific programs, and layered products.
- Modify or display VAXcluster parameters using CONFIGURATION commands.
- Load system services using SYS\_LOADABLE commands, which add and remove executive loaded images from the set of images loaded at boot time.

The following sections discuss different SYSMAN tasks, including

- Defining a management environment
- Centralizing system management
- Adjusting your profile
- Executing commands and command procedures
- Creating and maintaining disk quotas
- Loading and unloading licenses
- Using Auto-Login Facility (ALF) functions
- Executing commands on remote nodes

# SYSMAN Description

1

## Defining a Management Environment

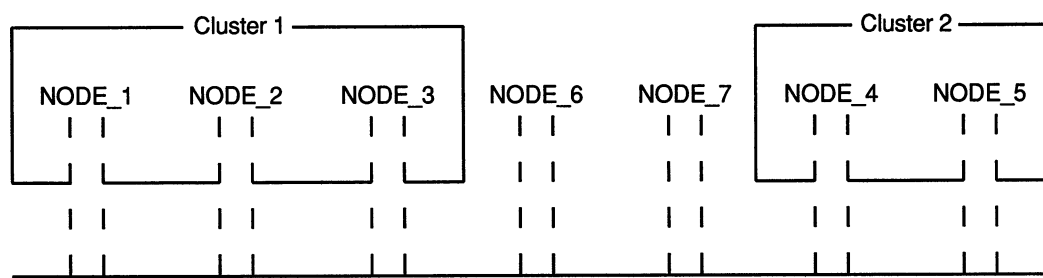
The default system management environment for SYSMAN is the local node from which you execute the utility.

Your first step in using SYSMAN is to define the target environment. The management environment can be the single node, group of nodes, or cluster you need to manage. You can define a management environment to be any of the following:

- Your local node
- Your own cluster
- A subset of nodes in your cluster
- A nonclustered node available through DECnet
- Another cluster
- A subset of nodes in another cluster
- Any group of individual nodes

Refer to Figure SM-1 during the following discussion of management environments. The figure depicts seven nodes, five of which are part of clusters.

Figure SM-1 Defining a Management Environment



Legend:

- == The DECnet Interconnect
- | | DECnet Connections
- Cluster Boundaries

ZK-6617-GE

If you execute SYSMAN from NODE\_1, then NODE\_1 is the management environment when SYSMAN starts. NODE\_1 is the **local** node because it is the node on which you invoked SYSMAN. All other nodes are **remote**. You can use NODE\_1 as the management environment, or you can change the environment to be any node, group of nodes, or cluster shown in Figure SM-1.

## 1.1 Defining Another Node as the Environment

Instead of managing the local node, you can define a management environment to be any node available through DECnet. For example, the following SET ENVIRONMENT command defines the management environment to be NODE\_2:

```
$ RUN SYSS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/NODE=NODE_2
%SYSMAN-I-ENV, current command environment:
    Individual nodes: NODE_2
    Username ALEXIS will be used on nonlocal nodes
```

A SYSMAN environment remains in effect until you change it or exit from the utility. Whenever you redefine an environment, SYSMAN displays the new context. You can always verify the current environment with the command SHOW ENVIRONMENT.

A management environment can also be a group of nodes. The following command defines the management environment to be NODE\_3, NODE\_6, and NODE\_7:

```
SYSMAN> SET ENVIRONMENT/NODE=(NODE_3,NODE_6,NODE_7)
Remote Password:

%SYSMAN-I-ENV, Current Command Environment:
    Individual nodes: NODE_6,NODE_7,NODE_3
    At least one node is not in local cluster
    Username ALEXIS will be used on nonlocal nodes
```

Unless you are working on your local node or within your own cluster, you are working in a **nonlocal** environment in SYSMAN. When defining a nonlocal environment, SYSMAN prompts for a password.

The utility also prompts for a password when you attempt to manage a system under a different user name. You can change your user name by including the /USERNAME qualifier on the SET ENVIRONMENT command.

The SET ENVIRONMENT/NODE command accepts logical names as values. Before you define logical names to use with SYSMAN, you must create the logical name table SYSMAN\$NODE\_TABLE as follows:

```
$ CREATE/NAME_TABLE/PARENT=LNMS$SYSTEM_DIRECTORY SYSMAN$NODE_TABLE
```

You might want to include this command in the file SYSMAN\$MANAGER:SYLOGICALS.COM, which is executed during system startup.

The following example demonstrates how you can define logical names to be a node or list of nodes and how to use those logical names to define the SYSMAN environment:

```
$ DEFINE FORTRAN ALICE,BILL,DIRK/TABLE=SYSMAN$NODE_TABLE
$ DEFINE PASCAL DIRK,JILL,CI_NODES/TABLE=SYSMAN$NODE_TABLE
$ DEFINE CI_NODES SYS2,SYS8/TABLE=SYSMAN$NODE_TABLE
$ RUN SYSS$SYSTEM SYSMAN
SYSMAN> SET ENVIRONMENT/NODE=(FORTRAN,PASCAL)
Remote Password:
```

# SYSMAN Description

```
%SYSMAN-I-ENV, current command environment:  
  Individual nodes: ALICE,BILL,DIRK,JILL,SYS2,SYS8  
  At least one node is not in the local cluster.  
  Username SYSTEM      will be used on nonlocal nodes.
```

Defining logical names is useful when you want to organize the nodes in your cluster according to specific categories (for example, all CI-based nodes or all nodes with FORTRAN installed).

## 1.2 Defining a Cluster Environment

To define a cluster as your target environment, use the /CLUSTER qualifier to the SET ENVIRONMENT command. There are two types of cluster environments:

- A local cluster
- A nonlocal cluster

The local cluster is the one where you are executing SYSMAN. For example, to expand the management environment from NODE\_1 to Cluster 1, enter the following command:

```
SYSMAN> SET ENVIRONMENT/CLUSTER  
%SYSMAN-I-ENV, Current Command Environment:  
  Clusterwide on local cluster  
  Username ALEXIS    will be used on nonlocal nodes
```

In the cluster environment shown in Figure SM-1, SYSMAN executes commands on all nodes in Cluster 1, namely NODE\_1, NODE\_2, and NODE\_3.

To manage a nonlocal cluster with SYSMAN, you must use the /NODE qualifier to identify the cluster you want to manage. If a cluster alias is defined for a cluster, the /NODE qualifier can include the alias rather than the node name. For example, to perform management tasks on Cluster 2 in Figure SM-1, you would enter another SET ENVIRONMENT command with the /CLUSTER qualifier and name one node within Cluster 2 using the /NODE qualifier. For example:

```
SYSMAN> SET ENVIRONMENT/CLUSTER/NODE=NODE_4  
Remote Password: (password)  
%SYSMAN-I-ENV, Current Command Environment:  
  Clusterwide on remote node NODE_4  
  Username ALEXIS    will be used on nonlocal nodes
```

When a specified node is not in a cluster, the /CLUSTER qualifier has no effect.

## 2 Centralizing System Management

After defining a management environment, you use SYSMAN to perform traditional system management tasks. When you enter commands on your local node, SYSMAN executes the commands in the target environment.

In cluster environments or any environment with multiple nodes, you enter a set of commands once and SYSMAN executes the commands sequentially on every node in the environment. SYSMAN displays the name of each node as it executes commands. For example, with SYSMAN you can run the INSTALL utility and make a file known on all nodes in the cluster by entering the following commands from your local node:

```
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> SET PROFILE/PRIVILEGE=CMKRNL
SYSMAN> DO INSTALL ADD/OPEN/SHARED WORK4:[CENTRAL]STATSHR
.
.
.
%SYSMAN-I-OUTPUT, Command execution on node PHILCO
%SYSMAN-I-OUTPUT, Command execution on node CEDAR
```

When a command does not execute successfully, SYSMAN displays an error message. If a node cannot execute a command or if a node does not respond within a given time period, SYSMAN displays a message before proceeding to the next node in the environment. A timeout period is optional, and can be established with the SET TIMEOUT command.

---

## 3 Adjusting Your Profile

Before SYSMAN executes any commands, it verifies that you are an authorized user and grants privileges, rights, and defaults. SYSMAN does this in one of the following two ways:

- If the cluster has common SYSUAF and RIGHTSLIST databases, SYSMAN checks to see whether the following conditions exist:
  - The node where the commands are being executed is part of the same cluster as the node where the commands are being entered.
  - The device name and file identification for the SYSUAF file match on the two nodes.
  - The device name and file identification for the RIGHTSLIST file match on the two nodes.

If these conditions are met, SYSMAN assigns the privileges, defaults, and identifiers in effect on the local node to an existing process called SMISERVER on each remote node.

- SYSMAN checks the information contained in the user authorization file (UAF) on each node in the environment to see that you are an authorized user. When you define an environment that is not part of your local cluster or when you change your user name, SYSMAN prompts for a password before checking the UAF. If you are an authorized user, SYSMAN assigns the privileges, defaults, and identifiers that it copies from your UAF record on that node to an existing process on the node called SMISERVER.

The utility uses a SMISERVER process on all nodes but the local node (from which you are running SYSMAN) to execute commands in an environment. The attributes of the SMISERVER process include a default device and directory, any rights identifiers defined in the UAF record, and a set of privileges. The attributes do not include symbolic names, logical

# SYSMAN Description

names, preset terminal characteristics, or key definitions established through a login command procedure. (The only environment that has the attributes defined in a login command procedure is the local node from which you are executing SYSMAN.)

SYSMAN allows you to modify the following attributes of the SMISERVER process:

- Current privileges
- Default device and directory
- DCL verification of DO commands

## 3.1 Modifying Your Current Privileges

Frequently, system management commands require enhanced privileges. You might need to add privileges before you execute certain commands in an environment. Typically, system managers have the same privileges on all nodes. However, if you do not have the required privileges on a node, SYSMAN cannot execute the command and returns an error message.

The SYSMAN command SET PROFILE temporarily changes your current privileges in an environment. For example, creating a quota file requires the SYSPRV privilege. Before you create a quota file, make SYSPRV (assuming you are authorized) one of your current privileges.

```
SYSMAN> SET PROFILE/PRIVILEGES=SYSPRV
SYSMAN> SHOW PROFILE
%SYSMAN-I-DEFDIR, Default directory on node BIKES -- WORK1:[MAEW]
%SYSMAN-I-DEFPRIV, Process privileges on node BIKES --
    TMPMBX
    OPER
    NETMBX
    SYSPRV
```

This profile is in effect until you change it, reset the environment, or exit from the utility.

The SET PROFILE command temporarily changes the attributes of your current local process. However, when you exit from the utility, all attributes are restored to the values that were current when SYSMAN was invoked.

## 3.2 Resetting the Default Device and Directory

Most often, the default device and directory specified in your UAF record is a first-level directory under which you create and maintain files and subdirectories. SYSMAN uses the default device and directory name in the current environment when resolving file specifications. It also assigns the default device and directory name to any files that you create during a session.

In some cases, you may need to reset the default device and directory in your SYSMAN profile to reflect the device and directory where you need to work. For example, you may have a directory containing command



procedures as well as some system management utilities that require the default directory be SYS\$SYSTEM.

Use the SET PROFILE/DEFAULT command to reset the default device and directory specification for your process and all server processes in the environment, as follows:

```
SYSMAN> SET PROFILE/DEFAULT=DMA1:[SMITH.COM]
```

The default device and directory specification remain in effect until you change it, reset the environment, or exit.

## 3.3 Setting DCL Verification

As described in Section 4.1, SYSMAN can execute DCL commands using the DO command. By default, SYSMAN DCL verification is turned off (SYSMAN does not display DCL command lines and data lines as they execute). Using the SET PROFILE/VERIFY command, you can turn on DCL verification. For example:

```
$ MCR SYSMAN
SYSMAN> DO SHOW USERS
%SYSMAN-I-OUTPUT, command execution on node CLACK
      VAX/VMS User Processes at 19-JUN-1990 13:32:19.45
      Total number of users = 1, number of processes =2
      Username      Interactive Subprocess Batch
      SYSTEM        1          1
SYSMAN> SET PROFILE/VERIFY
SYSMAN> DO SHOW USERS
%SYSMAN-I-OUTPUT, command execution on node CLACK
SETNOON
DO SHOW USERS
      VAX/VMS User Processes at 19-JUN-1990 13:32:19.45
      Total number of users = 1, number of processes =2
      Username      Interactive Subprocess Batch
      SYSTEM        1          1
```

In this example, the second SHOW USERS command is displayed because DCL verification was turned on with the SET PROFILE/VERIFY command.

## 4 Executing Commands

SYSMAN lets you gain access to management environments quickly and perform system management tasks as simply as possible. The SYSMAN commands SET ENVIRONMENT and SET PROFILE define the management sphere and set up privileges that subsequent tasks require.

To perform these management tasks, you use DCL commands and SYSMAN commands. The following sections describe the types of commands available to you and the different ways in which you can execute them.

# SYSMAN Description

---

## 4.1 DCL Commands

The SYSMAN command DO executes a DCL command or DCL command procedure on all nodes in an environment. Each DO command executes as an independent subprocess, so there is no process context retained between DO commands. For this reason, you must express all DCL commands in a single command string, and you cannot run a procedure that requires input.

When operating in a cluster environment, SYSMAN executes DO commands sequentially on all nodes in the cluster. After a command completes or times out on one node, SYSMAN sends it to the next node in the environment. Any node that is unable to execute a command returns an error message.

Some DCL commands, such as MOUNT/CLUSTER or SET QUORUM /CLUSTER, operate clusterwide by design. To avoid executing such commands excessively in SYSMAN, define the environment to be a single node within the cluster.

---

## 4.2 SYSMAN Command Procedures

The SYSMAN command @ executes SYSMAN command procedures on each node in the environment. For example, you could create a SYSMAN command procedure to display the current date and system time for each cluster node as follows:

```
$ CREATE TIME.COM
SET ENVIRONMENT/CLUSTER
CONFIGURATION SHOW TIME [Ctrl/Z]
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> @TIME
%SYSMAN-I-ENV, Current command environment:
      Clusterwide on local cluster
      Username SYSTEM will be used on nonlocal nodes

System time on node ZENITH: 19-JUN-1990 13:32:19.45
System time on node HOSTA: 19-JUN-1990 13:32:27.79
System time on node KEBBI: 19-JUN-1990 13:32:58.66
SYSMAN>
```

---

## 4.3 DISKQUOTA Commands

DISKQUOTA commands allow you to control disk usage. You can assign disk quotas to users and maintain an accurate record of disk use for ODS Level 2 disks. You create a quota file for each disk *except the system disk*. The quota file records the current usage and the maximum disk usage for all users.

SYSMAN allows you to access disks that are normally unavailable from your local node. With SYSMAN you can obtain a display of all disks on the other nodes, including those that are mounted privately or used as system disks. You can run DISKQUOTA on any available disk without logging in to each node.

## 4.3.1 Contents of a Quota File

The first step in allocating disk space is to create a quota file for each volume or each volume set. A quota file records all users allowed to use the disk and shows their current disk usage as well as their maximum disk allocation. A quota file has the following format:

```
SYSMAN> DISKQUOTA SHOW * /DEVICE=DJA$14:

    UIC ①      Usage ②      Permanent Quota ③      Overdraft Limit ④
[0,0]          0          333333              3333
[TTD,DAVIS]   15590         333333              3333
[TTD,MORGAN]  1929          333333              3333
[MKT,MORSE]   7650          333333              3333
.
.
```

- ① User Identification Code (UIC) of each user entitled to maintain files on the volume. UIC [0,0] appears in all quota files; use it as a template to set default values for quotas and overdrafts.
- ② Number of disk blocks currently dedicated to a user's files. This number includes the blocks allocated (shown by the DCL command DIRECTORY /SIZE=ALL /BY\_OWNER=owner), plus at least one block in the index file for every file owned by the user.
- ③ Maximum number of blocks on the volume that a user's files can occupy. Once exceeded, the system issues an error message when a file is created.
- ④ Number of blocks by which a user can exceed the quota.

The absolute maximum number of blocks permitted a user on a volume is the sum of the quota and the overdraft. Only users with the EXQUOTA privilege can bypass disk quota restrictions.

A quota file, which is called QUOTA.SYS, is stored in directory [000000] with other system files and requires one block of disk storage for every 16 entries.

## 4.3.2 Establishing Disk Quotas

To create a quota file on a disk, use the command DISKQUOTA CREATE and specify the target disk with the /DEVICE qualifier. The command sets up a file called QUOTA.SYS in directory [000000].

The quota file has one entry, UIC [0,0], that is used to store default values for quotas and overdrafts. After creating the file, use the DISKQUOTA MODIFY command to adjust [0,0] to an appropriate value for the device.

**Note:** If you create a quota file or enable disk quotas on a disk that has files on it, use the DISKQUOTA REBUILD command to update the disk quota entries with the current usage information.

# SYSMAN Description

---

## 4.3.3 Quota Enforcement

During normal use of a volume with a quota file, the system automatically updates the usage counts as users create, delete, extend, and truncate files.

To create new files, usage counts must be below the permanent quota. If an operation to add new files exceeds the permanent quota, the system prohibits the operation and issues an error message.

Even if you exceed the quota, a current file can be extended by entering the operation a second time. Operations to extend the file succeed until the usage exceeds the sum of the permanent quota and the overdraft values.

The DISKQUOTA DISABLE command suspends quota operations on a volume. To permanently disable quotas on a device, disable the quotas with the DISKQUOTA DISABLE command and delete the file QUOTA.SYS. Otherwise, the system implicitly enables quotas when the disk is mounted. Whenever quotas are enabled on a volume—either implicitly with the MOUNT command or explicitly with the DISKQUOTA ENABLE command—you must update disk quota information using the command DISKQUOTA REBUILD. In updating the quota file, the system adds new UICs and corrects usage counts for each user.

---

## 4.4 Cluster Parameter Commands

The CONFIGURATION commands let you inspect, set, or modify certain cluster parameters, specifically system time and the security data of a local area cluster.

A local area cluster has a group number and password that ensures its integrity on an Ethernet. This security data identifies which nodes belong to the local area cluster, thus preventing access to a cluster by an unauthorized node. The CONFIGURATION command set allows you to display and modify the group number and password.

---

## 4.5 System Parameter Commands

The PARAMETERS commands let you inspect, set, and modify system parameters and parameter files in the current environment. You can modify a standard system parameter file for use in a subsequent system boot, or dynamically modify the parameter values of the active system (dynamic parameters only). The PARAMETERS commands duplicate the parameter functions of the VMS System Generation Utility (SYSGEN).

**Caution:** Many SYSGEN parameters affect other parameters and the performance of the system. Digital recommends that you modify system parameters by editing the file SYS\$SYSTEM:MODPARAMS.DAT and invoking AUTOGEN.

The procedure for inspecting or modifying a system parameter file is summarized as follows:

- 1 Use the **PARAMETERS SHOW** command to examine the parameter values in the work area.
- 2 Enter a **PARAMETERS USE** command to initialize the work area to use the values of a new parameter file, the current system parameter file, or the default values, if the active values do not provide a suitable base for subsequent operations.
- 3 Enter **PARAMETER SET** commands to modify parameters on an individual basis. These modifications have no effect outside the work area.
- 4 Enter a **PARAMETERS WRITE** command to create a parameter file, modify the current system parameter file on disk, or modify the active system in memory (dynamic parameters only).

See the *Guide to Setting Up a VMS System* for further information on inspecting and modifying system parameters.

---

## 4.6 Startup Commands

The **STARTUP** command set displays and modifies the execution characteristics of command procedures and executable programs that start the VMS operating system, site-specific programs, and layered products. Using **STARTUP** commands you can display the names of startup files, modify their characteristics, and add site-specific files or layered products.

The command procedure **STARTUP.COM** calls all files involved in startup tasks. These startup files reside in **SYS\$STARTUP**. A file with the logical name **STARTUP\$STARTUP\_VMS** lists the names of files that start the VMS operating system and describes the manner in which they execute. A file with the logical name **STARTUP\$STARTUP\_LAYERED** lists files that start site-specific products and layered products.

**STARTUP\$STARTUP\_VMS** is provided for your information. You can display it, but Digital recommends that you do not modify it. However, you might want to modify **STARTUP\$STARTUP\_LAYERED** so that it contains all the files you want to execute at your installation. The only way to modify the **STARTUP\$STARTUP\_LAYERED** is with the **STARTUP** command set.

**STARTUP\$STARTUP\_LAYERED** lists the files and command procedures that start site-specific products and layered products. It contains the following characteristics of each file:

- Name of the component file to be run. The file type must be either **EXE** or **COM**.
- Phase in which the component file is to run. You can install layered products during one of four phases. Each phase describes a minimum

## SYSMAN Description

environment that exists at that point in the startup process, as follows:

- 1 LPBEGIN—SYSTARTUP\_V5.COM executes here as well as any other files that prepare an environment necessary for layered products.
  - 2 LPMAIN—The majority of layered products execute here.
  - 3 LPBETA—Layered products that have a dependency on previously installed products execute here.
  - 4 END—Products that are dependent on layered products execute here.
- Mode by which the component file is to run. A command procedure or image can be run directly, or it can be spawned. A command procedure can be run in batch.
  - Node restrictions for the component. This is either the node or nodes on which the component file should be run, or the node or nodes on which the component file should *not* be run.
  - Parameters passed to the component file for execution. You can pass up to eight parameters, using the following format:

(P1:args,P2:args,...)

See the *Guide to Setting Up a VMS System* for more information on the system startup procedure.

### 4.7 Loadable Image Commands

**Caution:** **SYS\_LOADABLE** commands are not intended for general use. Only advanced system programmers should use these commands.

You can use **SYS\_LOADABLE** commands to add or remove executive loaded images from the list of images that the **SYSDINIT** process loads when the system boots. Use these commands in conjunction with any routines or special algorithms you have written. Your routines and algorithms can implement site-specific policies, replace built-in VMS policies, or augment built-in VMS policies.

See the *VMS Version 5.4 New Features Manual* for more information on creating site-specific loadable images and including them within VMS.

### 4.8 LICENSE Commands

Using the **SYSMAN** commands **LICENSE LOAD** and **LICENSE UNLOAD**, you can load and unload licenses on multiple systems and on nonlocal systems in the management environment. To use these commands, you need the following privileges:

- **CMKRNL** (change mode to kernel)
- **SYSNAM** (create system logical names)

- SYSPRV (system privileges)

These commands are a subset of the License Management Facility (LMF) commands. For more information about the LMF, see the *VMS License Management Utility Manual*.

## 4.9 Auto-Login Facility (ALF) Commands

Using the SYSMAN command ALF ADD, you can use the Auto-Login Facility (ALF) to associate a terminal or port with a particular user name. By doing this, you can allow selected users to log in to certain terminals without specifying a user name. For more information about ALF, see the *Guide to VMS System Security*.

## 4.10 SPAWN and ATTACH Commands

The SPAWN command creates a subprocess of your current process. The ATTACH command transfers control from your current process to the specified process.

Once in SYSMAN, you can use the SPAWN command to leave temporarily and perform other functions (such as displaying a directory listing or printing a file) and then return to SYSMAN.

Note that SPAWN performs actions on the local node only. If you want to execute DCL commands or command procedures throughout your environment, use the DO command.

For example, to leave SYSMAN to read MAIL, enter the following commands:

```
SYSMAN> SPAWN
$ MAIL
.
.
.
$ LOGOUT
Process SYSTEM_1 logged out at 28-JUN-1990 10:05:17.24
SYSMAN>
```

Using the ATTACH command, you can transfer control from your current process to another process in your job.

For example, the following command transfers control from the current process to a subprocess named SMYTH\_3:

```
SYSMAN> ATTACH SMYTH_3
%DCL-S-RETURNED, control returned to process SMYTH_3
```

# SYSMAN Description

## 4.11 Defining Keys to Execute SYSMAN Commands

As an alternative to typing lengthy command lines, the SYSMAN Utility allows you to define keys to execute SYSMAN commands. For example, you can define a key to execute a SET/ENVIRONMENT command as follows:

```
SYSMAN> DEFINE/KEY/TERMINATE
_Key name: KP0
_Translation: "SET ENVIRONMENT/NODE=(BARBOR,TWILL,MOOR,TORIN)"
```

Once the key is defined, you can press keypad 0, and SYSMAN executes the SET ENVIRONMENT command. Note that any keys that you define in SYSMAN lose their meaning when you exit from SYSMAN, unless you define them in the SYSMAN initialization file (see Section 4.12).

## 4.12 Executing Commands from an Initialization File

You can create an initialization file that SYSMAN will use each time you invoke the utility. In the SYSMAN initialization file, you can perform tasks such as defining keys, or setting the SYSMAN environment.

The default file specification for the SYSMAN initialization file is SYS\$LOGIN:SYSMANINI.INI. If you want your SYSMAN initialization file to have a different file specification, you must define the logical name SYSMANINI to point to the location of the file.

The following is a sample initialization file in which several keys are defined:

```
$ TYPE SYSMANINI.INI
DEFINE/KEY/TERMINATE KP0 "SET ENVIRONMENT/CLUSTER/NODE=(BARBOR,TWILL,MOOR,TORIN)"
DEFINE/KEY/TERMINATE KP1 "CONFIGURATION SHOW TIME"
DEFINE/KEY/TERMINATE KP2 "SHOW PROFILE"
.
.
.
```

## 4.13 Executing Commands on Remote Nodes

SMISERVER is the detached process responsible for executing SYSMAN commands on remote nodes.

A workstation that is not part of a cluster will not start SMISERVER automatically. If you want to start SMISERVER on a standalone workstation, insert the following command in the site-specific startup procedure SYS\$MANAGER:SYSTARTUP\_V5.COM:

```
$ @SYS$SYSTEM:STARTUP SMISERVER
```

You can also use this command to restart SMISERVER without rebooting the system.

On any node that is part of a cluster, SMISERVER is normally started by the system startup procedure SYS\$SYSTEM:STARTUP.COM. (The system must have a SYSGEN parameter VAXCLUSTER with a value of 1 or more.)



## 5

### Command Summary

This section summarizes the SYSMAN commands by function and format. See the Command Section for a complete description.

| Task   | Command                                  | Function  |
|--|--|---|
| Running a command procedure containing SYSMAN commands           | @  | Requests that SYSMAN read subsequent command input from the specified file or device.           |
| Using the functions of the Auto-Login Facility (ALF)             | ALF ADD                                  | Adds a record to the ALF database.  |
|  | ALF REMOVE                               | Deletes one or more records from the ALF database.  |
|  | ALF SHOW                                 | Displays one or more records in the ALF database  |
| Transferring control from the current process to another process | ATTACH                                   | Permits you to transfer control from your current process to the specified process in your job. |
| Inspecting and modifying cluster parameters                      | CONFIGURATION SET CLUSTER_AUTHORIZATION  | Updates security data in a local area cluster.  |
|  | CONFIGURATION SET TIME                   | Updates system time.  |
|  | CONFIGURATION SHOW CLUSTER_AUTHORIZATION | Displays cluster security data.   |
|  | CONFIGURATION SHOW TIME                  | Displays current system time.   |
| Defining keys  | DEFINE/KEY                               | Defines a key to execute a SYSMAN command   |
|  | SHOW KEY                                 | Displays key definitions  |
| Controlling and monitoring disk usage                            | DISKQUOTA ADD                            | Adds an entry to a disk quota file.   |
|  | DISKQUOTA CREATE                         | Creates and enables a disk quota file for a volume that does not contain one.                   |
|  | DISKQUOTA DISABLE                        | Suspends disk quota operations on a volume.   |
|  | DISKQUOTA ENABLE                         | Resumes disk quota operations on a volume.  |
|  | DISKQUOTA MODIFY                         | Changes an entry in the quota file or adjusts the default quota and overdraft values.           |
|  | DISKQUOTA REBUILD                        | Reconstructs the disk usage counts for all entries.   |
| Using DCL commands   | DISKQUOTA REMOVE                         | Removes an entry from a disk quota file.  |
|  | DISKQUOTA SHOW                           | Displays disk quotas and usage counts.  |
| Manipulating licenses  | DO                                       | Executes a DCL command or DCL command procedure.  |
|  | LICENSE LOAD                             | Activates a license that is registered in the LICENSE database.                                 |
|  | LICENSE UNLOAD                           | Deactivates a license that is registered in the LICENSE database.                               |

# SYSMAN Description

| Task  | Command                   | Function   |
|---|---------------------------|--|
| Inspecting and modifying system parameters        | PARAMETERS USE            | Initializes the current work area with system parameter values and the name of the site-independent command procedure.   |
|   | PARAMETERS SHOW           | Displays the values of system parameters in the work area, plus the default, minimum, and maximum values of the parameters and their units of measure.   |
|   | PARAMETERS SET            | Modifies the value of a system parameter in the work area.   |
|   | PARAMETERS WRITE          | Writes the system parameter values and the name of the site-independent command procedure from the work area to a parameter file, the current system parameter file, or the active system in memory. |
|   | PARAMETERS ENABLE CHECKS  | Ensures that range checks are in effect. Enables range checks after a PARAMETERS DISABLE CHECKS command.   |
|   | PARAMETERS DISABLE CHECKS | Inhibits range checks on system parameter values specified in a PARAMETERS SET command.  |
| Using SYSMAN                                      | SET ENVIRONMENT           | Establishes a management context for subsequent SYSMAN commands.   |
|   | SET PROFILE               | Modifies the default device and directory and the current privileges for the current management environment, and allows you to set DCL verification for future DO commands.                          |
|   | SET TIMEOUT               | Establishes the amount of time that SYSMAN waits for a node to respond.  |
|   | EXIT                      | Terminates the SYSMAN session and returns control to the DCL command level.  |
|   | HELP                      | Provides information on SYSMAN commands.   |
|   | SHOW ENVIRONMENT          | Displays the current command context.  |
|   | SHOW PROFILE              | Displays the default device and directory and the current privileges.  |
| Leaving SYSMAN temporarily to perform other tasks | SHOW TIMEOUT              | Displays the current timeout period.   |
|   | SPAWN                     | Creates a subprocess of the current process, where the context of the subprocess is copied from the current process.   |

## SYSMAN Description

| <b>Task</b>   | <b>Command</b>                       | <b>Function</b>  |
|---|--------------------------------------|--|
| Modifying the startup database or program characteristics | STARTUP ADD                          | Adds an executable file or command procedure to the startup database.                                  |
|   | STARTUP DISABLE                      | Prevents a component of the startup database from executing on one or more nodes in the environment.   |
|   | STARTUP ENABLE                       | Allows a component of the startup database to execute.   |
|   | STARTUP MODIFY                       | Edits a record in the startup database describing how a startup component executes.                    |
|   | STARTUP REMOVE                       | Removes one or more components from the startup database.  |
|   | STARTUP SET DATABASE<br>STARTUP SHOW | Determines the default database.<br>Displays the name of the current startup database or its contents. |
| Manipulating executive loaded images                      | SYS_LOADABLE ADD                     | Adds an executive loaded image to the set of images loaded at boot time.                               |
|   | SYS_LOADABLE REMOVE                  | Removes an executive loaded image from the set of images loaded at boot time.                          |

---

## SYSMAN Usage Summary

The VMS System Management Utility (SYSMAN) centralizes system management, enabling you to manage nodes or clusters from one location.

---

### FORMAT                    RUN SYS\$SYSTEM:SYSMAN

#### Parameters

*None.*

---

#### usage summary

To invoke SYSMAN, enter the following command at the DCL prompt:

```
$ RUN SYS$SYSTEM:SYSMAN
```

The utility displays the following prompt:

```
SYSMAN>
```

You can then enter SYSMAN commands. These commands follow the standard rules of DCL syntax.

As an alternative, you can enter the RSX command MCR, which expands to RUN SYS\$SYSTEM:

```
$ MCR SYSMAN
```

With the MCR command, you can invoke SYSMAN and supply a command in one command string. With any SYSMAN command (except SET ENVIRONMENT), SYSMAN executes the command string and exits. After executing a SET ENVIRONMENT command, the utility returns the SYSMAN> prompt.

To exit from SYSMAN and return to the DCL command level, enter the EXIT command at the SYSMAN> prompt or press Ctrl/Z.

**Note: SYSMAN has the following restrictions:**

- **To use SYSMAN, you must have the OPER privilege on the local node and authorization for the OPER or SETPRV privilege on any remote node in the management environment.**

**You must also have the privileges required by individual commands, as documented in the Command Section. To determine which privileges are required for DCL commands or for system management utilities, refer to the *VMS DCL Dictionary* or the appropriate utility reference manual.**

- **You cannot run SYSMAN from a batch job in any environment that requires a password.**

## SYSMAN Usage Summary

- **Some DCL commands, such as SET CLUSTER/QUORUM, MOUNT/CLUSTER, and some forms of the REPLY command, operate clusterwide by design, and should not be run in a SYSMAN environment defined as a cluster.**

# **SYSMAN Usage Summary**

---

## **SYSMAN COMMANDS**

This section describes the SYSMAN commands and illustrates their use.

---

## @ (Execute Procedure)

Requests that SYSMAN read subsequent command input from the specific file or device.

---

**FORMAT**            @ *filespec*

---

**PARAMETER**        *filespec*

Specifies either the input device or the command procedure you want to execute. The default file type is .COM. You cannot use a wildcard characters in the file specification.

---

### DESCRIPTION

Use the @ command to execute a command procedure containing SYSMAN commands. To execute the command procedure, invoke SYSMAN, place the @ command at the beginning of a command line, then specify the file name of the command procedure.

The command procedure can contain any valid SYSMAN command.

---

### EXAMPLES

```

1  $ CREATE ENV.COM
    SET ENVIRONMENT
    SHOW PROFILE [Ctrl/Z]
    $ MCR SYSMAN
    SYSMAN> @ENV.COM
    %SYSMAN-I-DEFDIR, default directory on node -- SYS$SYSROOT:[SYSMGR]
    %SYSMAN-I-DEFPRIV, default process privileges on node --
        CMKRNL
        CMEXEC
        SYSNAM
        .
        .
        .
        GRPPRV
        READALL
        SECURITY
    SYSMAN>
  
```

This example shows how to create a command procedure that sets the SYSMAN environment to the local node, and displays the current profile. These commands execute when you enter the @ENV.COM command.

# **SYSMAN**

## **@ (Execute Procedure)**

```
2 $ CREATE TIME.COM
  SET ENVIRONMENT/CLUSTER
  CONFIGURATION SHOW TIME Ctrl/Z
  $ MCR SYSMAN
  SYSMAN> @TIME
System time on node ZENITH: 19-JUN-1990 13:32:19.45
System time on node HOSTA: 19-JUN-1990 13:32:27.79
System time on node KEBBI: 19-JUN-1990 13:32:58.66
```

**This example shows how to create and execute a command procedure that shows the current date and time for all the nodes in the cluster.**



---

## ALF ADD

Adds a new record to the Auto-Login Facility (ALF) database.

Requires read (R) and write (W) access to the SYSALF database (SYS\$SYSTEM:SYSALF.DAT by default).

---

**FORMAT**            **ALF ADD** *device user*

---

### PARAMETERS

#### ***device***

Specifies the terminal name or port name that you want to assign to a user name. The parameter **device** must be a terminal name if you do not specify qualifiers on the command line, and can be either a logical name or an abbreviated device name.

#### ***user***

Specifies the user name of the account that you want to assign to a particular terminal or port.

---

### QUALIFIERS

#### ***/TERMINAL (default)***

Directs SYSMAN to check whether the device name you specified is a terminal on the target system. The parameter **device** can be a logical name or an abbreviated device name (which SYSMAN translates to a full device name).

#### ***/PORT***

Directs SYSMAN to check whether the device name you specified is a valid port. If the port name contains special characters, such as a slash (/), or if it contains lowercase letters that you want to preserve, you must enclose the port name within quotation marks (" ").

#### ***/PROXY***

Directs SYSMAN to check whether the device name you specified is a valid port. SYSMAN also checks that **device** is in the NODE::USERNAME format.

#### ***/LOG***

Directs SYSMAN to display the device names and user names as they are added to the ALF database.

---

### DESCRIPTION

You can use the SYSMAN ALF ADD command to associate a terminal or port with a particular user name. This will enable certain users to log in to certain terminals or ports without specifying a user name.

The SYSMAN ALF ADD command adds a new record to the ALF database.

# **SYSMAN**

## **ALF ADD**

---

### **EXAMPLE**

```
SYSMAN> ALF ADD TTA3 JBERGERON  
SYSMAN> ALF ADD "MN34C3/LC-1-2" FMARTIN /PORT
```

In this example, the first command assigns terminal TTA3 to user JBERGERON. The second command assigns port MN34C3/LC-1-2 to user FMARTIN.

---

## **ALF REMOVE**

Removes one or more records from the ALF database.

Requires read (R) and write (W) access to the SYSALF database (SYS\$SYSTEM:SYSALF.DAT).

---

**FORMAT**            **ALF REMOVE**    *device*

---

**PARAMETER**        *device*

Specifies the terminal name or port name whose record you want to remove from ALF. You can use wildcard characters in the terminal name or port name.

**Note:** When you specify device to remove a record from the ALF database, be sure to use the correct format. Enter the ALF SHOW command to display the device name format. Include special characters such as underscores (\_) and colons (:).

---

**QUALIFIERS**        ***/USERNAME=user***

Allows you to remove a record in ALF by specifying a user name rather than a terminal name or port name. You can use wildcard characters with the /USERNAME qualifier.

***/CONFIRM***

Causes SYSMAN to display a message asking you to verify that you want to remove the record.

***/LOG***

Causes SYSMAN to display each device name and user name after it has been removed from the ALF database.

---

**DESCRIPTION**      The SYSMAN ALF REMOVE command removes one or more records from the ALF database.

---

### **EXAMPLE**

```
SYSMAN> ALF REMOVE _TTA3:  
SYSMAN> ALF REMOVE /USERNAME=SMITHSON
```

In this example, the first command removes the record for terminal TTA3. The second command removes all records assigned to user name SMITHSON.

# SYSMAN

## ALF SHOW

---

## ALF SHOW

Displays one or more records in the ALF database.

Requires read (R) and write (W) access to the SYSALF database (SYS\$SYSTEM:SYSALF.DAT).

---

**FORMAT**            **ALF SHOW** *[device]*

---

**PARAMETER**        *[device]*

Specifies the terminal name or port name whose record you want to display. You can use wildcard characters in the terminal name or port name.

---

**QUALIFIERS**        **/USERNAME=user**

Allows you to display the records held by the specified user. You can use wildcard characters with this qualifier.

**/OUTPUT[=filespec]**

Allows you to direct the output of the command to a file. If you do not include a file specification with this qualifier, SYSMAN writes the output to the file SYSMAN.LIS in your default directory.

---

**DESCRIPTION**      The SYSMAN ALF SHOW command displays one or more records in the ALF database.

---

## EXAMPLE

```
SYSMAN> ALF SHOW TTA* /USERNAME=MANESS /OUTPUT=ALF.TXT
```

In this example, the records for all terminals named TTAx that are assigned to user MANESS, are selected and directed to the file ALF.TXT.

---

## **ATTACH**

Transfers control from your current process (which then hibernates) to the specified process in your job.

The ATTACH and SPAWN commands cannot be used if your terminal has an associated mailbox.

---

**FORMAT**            **ATTACH** *[process-name]*

---

**PARAMETER**        ***process-name***  
Specifies the name of a parent process or a spawned subprocess to which control passes. The process must already exist, be part of your current job tree, and share the same input stream as your current process. However, the process cannot be your current process or a subprocess created with the /NOWAIT qualifier.

Process names can contain from 1 to 15 alphanumeric characters. If a connection to the specified process cannot be made, an error message is displayed.

---

**QUALIFIERS**        ***/PARENT***  
Allows you to attach to your process' parent process. If there is no parent process, you receive an error message.

---

**DESCRIPTION**      The ATTACH command allows you to connect your input stream to another process. You can use the ATTACH command to change control from one subprocess to another subprocess or to the parent process.

When you enter the ATTACH command, the parent or "source" process is put into hibernation, and your input stream is connected to the specified destination process. You can use the ATTACH command to connect to a subprocess that is part of a current job (left hibernating as a result of the SPAWN/WAIT command or another ATTACH command) as long as the connection is valid. No connection can be made to the current process, to a process that is not part of the current job, or to a process that does not exist. If you attempt any of these connections, you receive an error message.

You can also use the ATTACH command in conjunction with the SPAWN /WAIT command to return to a parent process without terminating the created subprocess. See the description of the SPAWN command for more details.

# **SYSMAN**

## **ATTACH**

---

### **EXAMPLE**

```
$ SPAWN
%DCL-S-SPAWNED, process SYSTEM_1 spawned
%DCL-S-ATTACHED, terminal now attached to process SYSTEM_1
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> ATTACH SYSTEM
%DCL-S-RETURNED, control returned to process SYSTEM
$
```

In this example, the SPAWN command creates a subprocess (SYSTEM\_1). After you invoke SYSMAN and enter the ATTACH command, you transfer the terminal's control back to the parent process (SYSTEM).

---

## CONFIGURATION SET CLUSTER\_AUTHORIZATION

Modifies security data in a local area cluster.  
Requires SYSPRV privilege.

---

|               |  |
|---------------|--|
| <b>FORMAT</b> | <b>CONFIGURATION<br/>SET CLUSTER_AUTHORIZATION</b> |
|---------------|--|

---

|                   |              |
|-------------------|--------------|
| <b>PARAMETERS</b> | <i>None.</i> |
|-------------------|--------------|

---

|                   |  |
|-------------------|--|
| <b>QUALIFIERS</b> | <p><b><i>/GROUP_NUMBER=[n]</i></b><br/>Specifies the cluster group number that is recorded in SYS\$SYSTEM:CLUSTER_AUTHORIZE.DAT. A group number uniquely identifies each local area cluster configuration on a single Ethernet. This number must be in the range from 1 to 4095 or 61440 to 65535.</p> <p><b><i>/PASSWORD=password</i></b><br/>Specifies a password for cluster access. A password consists of 1 to 31 characters, including alphanumeric characters, the dollar sign, and underscore. A password provides a second level of validation to ensure the integrity of individual clusters on the same Ethernet that accidentally use identical group numbers. A password also prevents an intruder who discovers the group number from joining the cluster.</p> |
|-------------------|--|

---

|                    |  |
|--------------------|--|
| <b>DESCRIPTION</b> | <p>The CONFIGURATION SET CLUSTER_AUTHORIZATION command modifies the group number and password of a local area cluster, as recorded in SYS\$SYSTEM:CLUSTER_AUTHORIZE.DAT. If your configuration has multiple system disks, SYSMAN automatically updates each copy of CLUSTER_AUTHORIZE.DAT, provided the environment is defined as a cluster (SET ENVIRONMENT/CLUSTER). For more information about CLUSTER_AUTHORIZE.DAT, see the <i>VMS VAXcluster Manual</i>.</p> |
|--------------------|--|

**Caution:** If you change either the group number or the password, you must reboot the entire cluster.

The file CLUSTER\_AUTHORIZE.DAT is initialized during execution of CLUSTER\_CONFIG.COM and maintained through the SYSMAN Utility. Under normal conditions, you do not need to alter records in the CLUSTER\_AUTHORIZE.DAT file interactively. However, if you suspect a security breach, use the CONFIGURATION commands in SYSMAN to make the change.

# **SYSMAN**

## **CONFIGURATION SET CLUSTER\_AUTHORIZATION**

---

### **EXAMPLE**

```
SYSMAN> SET ENVIRONMENT/CLUSTER/NODE=ASCONA
SYSMAN> SET PROFILE /PRIVILEGE=SYSPRV
SYSMAN> CONFIGURATION SET CLUSTER_AUTHORIZATION/PASSWORD=GILLIAN
%SYSMAN-I-CAFOLDGROUP, existing group will not be changed
%SYSMAN-I-GRPNOCHG, Group number not changed
SYSMAN-I-CAFREBOOT, cluster authorization file updated.
The entire cluster should be rebooted.
```

The CONFIGURATION SET CLUSTER\_AUTHORIZATION command in this example sequence modifies the cluster password. Note that the environment is defined to be a cluster, and the SYSPRV privilege is established before entering the CONFIGURATION SET CLUSTER\_AUTHORIZATION command.



---

## CONFIGURATION SET TIME

Modifies the current system time.

Requires LOG\_IO privilege, and, in a cluster environment, SYSLCK privilege.

---

|               |                                      |
|---------------|--------------------------------------|
| <b>FORMAT</b> | <b>CONFIGURATION SET TIME [time]</b> |
|---------------|--------------------------------------|

---

|                   |              |
|-------------------|--------------|
| <b>PARAMETERS</b> | <i>None.</i> |
|-------------------|--------------|

---

|                    |  |
|--------------------|--|
| <b>DESCRIPTION</b> | The CONFIGURATION SET TIME command allows you to reset the system time. Specify a time value using the following format: |
|--------------------|--|

[dd-mmm-yyyy[:]] [hh:mm:ss.cc]

See the *VMS DCL Concepts Manual* for a discussion of acceptable time formats.

In an environment of individual nodes, SYSMAN sets the time to the specified value on each node. Without a time specification, SYSMAN sets the time according to the time-of-year clock on each node.

In a cluster environment, SYSMAN sets the time to the specified value on each node. However, if you do not specify a value, SYSMAN uses the time-of-year clock. In a local cluster, SYSMAN reads the clock on the node from which you are executing SYSMAN and assigns this value to all nodes in the cluster. In a remote cluster, SYSMAN reads the clock on the target node in the cluster and assigns that value to all nodes. Note that the time-of-year clock is optional for some processors; see the *VAX Hardware Handbook* for further information.

SYSMAN uses special processing in a cluster environment to ensure that all processors in the cluster are set to the same time. Because of communication and processing delays, it is not possible to synchronize clocks exactly. However, the variation is typically less than a few hundredths of a second. If SYSMAN cannot set the time to within one half second of the specified time, you receive a warning message that names the node that failed to respond quickly enough.

As a result of slight inaccuracies in each processor clock, times on various members of a cluster tend to drift apart. The following procedure synchronizes system times in a cluster environment:

```
$ SYNCH_CLOCKS:
$ RUN SYSSYSTEM:SYSMAN
    SET ENVIRONMENT/CLUSTER
    CONFIGURATION SET TIME
    EXIT
$ WAIT 6:00:00
$ GOTO SYNCH_CLOCKS
```

## **SYSMAN CONFIGURATION SET TIME**

The procedure sets the time on all cluster nodes to the value obtained from the local time-of-year clock, waits 6 hours, then resets the time for the cluster.

---

### **EXAMPLE**

```
SYSMAN> SET ENVIRONMENT/NODE= (ASCONA, LUGANO, LUCERN)  
SYSMAN> SET PROFILE /PRIVILEGE=LOG_IO  
SYSMAN> CONFIGURATION SET TIME 12:38:00
```

The CONFIGURATION SET command in this example sequence modifies the system time on nodes ASCONA, LUGANO, and LUCERN.

---

## CONFIGURATION SHOW CLUSTER\_AUTHORIZATION

Displays the group number and multicast address of a local area cluster.

Requires SYSPRV privilege.

---

**FORMAT**                    **CONFIGURATION**  
**SHOW CLUSTER\_AUTHORIZATION**

---

**PARAMETERS**    *None.*

---

**QUALIFIERS**    */OUTPUT[=filespec]*  
Redirects output from SYS\$OUTPUT to the specified file. If no file specification is provided, SYSMAN writes the output to SYSMAN.LIS in the current directory.

---

**DESCRIPTION**    The CONFIGURATION SHOW CLUSTER\_AUTHORIZATION command displays the group number and multicast address of a local area cluster, as recorded in SYS\$SYSTEM:CLUSTER\_AUTHORIZE.DAT during the CLUSTER\_CONFIG dialog. In a cluster or multinode environment, SYSMAN displays the group number of the first node and then displays the names of any nodes in the cluster whose group numbers, passwords, or both, are different. This command also displays the multicast address of the cluster.

---

### EXAMPLE

```
SYSMAN> SET ENVIRONMENT/CLUSTER/NODE=ZENITH
.
.
.
SYSMAN> SET PROFILE /PRIVILEGE=SYSPRV
SYSMAN> CONFIGURATION SHOW CLUSTER_AUTHORIZATION
Node ZENITH: Cluster group number 65240
Multicast address: AB-00-04-01-F2-FF
```

The CONFIGURATION SHOW CLUSTER\_AUTHORIZATION command in this example displays the group number and multicast address of node ZENITH. Because the group number and password on other nodes in the cluster are identical, no further information is displayed.

# **SYSMAN**

## **CONFIGURATION SHOW TIME**

---

# **CONFIGURATION SHOW TIME**

Displays the current date and system time to the hundredths of a second.

---

## **FORMAT            CONFIGURATION SHOW TIME**

---

**PARAMETERS**    *None.*

---

**QUALIFIERS**    ***/OUTPUT[=filespec]***  
Redirects output from SYS\$OUTPUT to the specified file. If no file specification is provided, SYSMAN writes the output to SYSMAN.LIS in the current directory.

---

## **EXAMPLE**

```
SYSMAN> SET ENVIRONMENT/CLUSTER/NODE=ZENITH
.
.
.
SYSMAN> CONFIGURATION SHOW TIME
System time on node ZENITH: 19-JUN-1990 13:32:19.45
System time on node HOSTA: 19-JUN-1990 13:32:27.79
System time on node KEBBI: 19-JUN-1990 13:32:58.66
```

The CONFIGURATION SHOW TIME command in this example displays the system time for all nodes in the cluster.

---

## DEFINE

Defines a key to execute a SYSMAN command. This enables you to press the key to enter a command, instead of typing the command name.

When you exit from SYSMAN, any SYSMAN key definitions you established will be lost unless you define them in the SYSMAN initialization file (see Section 4.12).

---

**FORMAT**      **DEFINE/KEY**    *key-name string*

---

**PARAMETERS**    *key-name*

Specifies the name of the key you are defining. Use the key names in Table SM-1, when defining keys.

**Table SM-1 Key Names in SYSMAN**

| Key name              | VT100      | VT52       | LK201             |
|-----------------------|------------|------------|-------------------|
| PF1                   | PF1        | red key    | PF1               |
| PF2                   | PF2        | blue key   | PF2               |
| PF3                   | PF3        | black key  | PF3               |
| PF4                   | PF4        | -          | PF4               |
| KP0, KP1-KP9          | keypad 0-9 | keypad 0-9 | keypad 0-9        |
| PERIOD                | period key | period key | period key        |
| COMMA                 | comma key  | comma key  | comma key         |
| MINUS                 | minus key  | minus key  | minus key         |
| ENTER                 | ENTER key  | ENTER key  | ENTER key         |
| UP, DOWN, LEFT, RIGHT | arrow keys | arrow keys | arrow keys        |
| FIND, INSERT_HERE     | -          | -          | Find, Insert Here |
| REMOVE, SELECT        | -          | -          | Remove, Select    |
| PREV_SCREEN           | -          | -          | Previous Screen   |
| NEXT_SCREEN           | -          | -          | Next Screen       |
| HELP, DO              | -          | -          | Help, Do          |
| F6-F10, F11-F14       | -          | -          | Function Keys     |
| F17-F20               | -          | -          | Function Keys     |

***string***

Specifies the string you want entered when you press the defined key. For example, you can define **string** as the SYSMAN command SHOW ENVIRONMENT or SHOW PROFILE.

# **SYSMAN**

## **DEFINE**

---

### **QUALIFIERS**

***/ECHO (default)***

***/NOECHO***

Specifies whether the command line is echoed after you press the defined key. Note that you cannot define a key using both the */NOECHO* and */NOTERMINATE* qualifiers.

***/IF\_STATE=state\_list***

***/NOIF\_STATE***

Specifies a list of states, any one of which must be set in order to enable the specified key definition. If you omit or negate this qualifier, the current state prevails.

***/LOCK\_STATE***

***/NOLOCK\_STATE (default)***

Retains the state specified by the */SET\_STATE* qualifier until you use the */SET\_STATE* qualifier again to change it.

***/SET\_STATE***

***/NOSET\_STATE***

Associates a state with the key you are defining. A state name can be any alphanumeric string. If you omit or negate this qualifier, the current state remains unchanged. You cannot define a key using both the */SET\_STATE* and */TERMINATE* qualifiers.

***/TERMINATE***

***/NOTERMINATE***

Determines whether the specified command string executes when you press the key. When you use */NOTERMINATE*, you must press the Return key to execute the command string. You cannot define a key using both the */SET\_STATE* and */TERMINATE* qualifiers.

---

### **DESCRIPTION**

The **DEFINE/KEY** command allows you to assign a key to a **SYSMAN** command. This enables you to execute the command by pressing the key. You can confirm which keys you have defined by using the **SHOW KEY** command.

When you exit from **SYSMAN**, any **SYSMAN** key definitions you established will be lost unless you define them in the **SYSMAN** initialization file (see Section 4.12).

---

### **EXAMPLES**

**I** `SYSMAN> DEFINE /KEY PF1 "SHOW PROFILE"`

This example shows how to define the keypad key PF1 as the **SYSMAN** command **SHOW PROFILE**. To execute the **SHOW PROFILE** command, press PF1 and then the Return key.

# **SYSMAN**

## **DEFINE**

2 SYSMAN> DEFINE /KEY KP0 /TERMINATE "CONFIGURATION SHOW TIME"

This example shows how to define the keypad key 0 as the CONFIGURATION SHOW TIME command. The /TERMINATE qualifier causes the SYSMAN command to execute when you press keypad key 0 without having to press Return.

---

## **DISKQUOTA ADD**

Adds an entry to a disk quota file and initializes the usage count to zero.  
Requires write (W) access to the quota file.

---

**FORMAT**            **DISKQUOTA ADD**    *uic*

---

**PARAMETER**        *uic*  
Specifies the user identification code (UIC) for which the quota entry is added. You can specify the UIC in numeric or alphanumeric format. For complete information on UIC specification, refer to the *VMS DCL Concepts Manual*.

You can also add quota entries for rights identifiers. These are rights granted a user with the AUTHORIZE Utility. Rights identifiers use an ID format rather than a UIC format. See the *VMS System Services Reference Manual* for a complete description.

When working in nonlocal environments, be careful that the alphanumeric UIC or rights identifiers that you use are valid for the environment.

---

**QUALIFIERS**        ***/DEVICE=device-spec***  
Specifies the location of the quota file. SYSMAN validates the device specification. You can specify a logical name for ***device-spec***. If you do, the logical name is translated in the target environment.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

***/OVERDRAFT=value***  
Specifies a positive integer that provides an overdraft value for the specified UIC. If omitted, the overdraft value defaults to the overdraft value in the entry for [0,0].

***/PERMQUOTA=value***  
Specifies a positive integer that provides the quota for the specified UIC. If omitted, the permanent quota defaults to the value of the quota in the entry for [0,0].

---

**DESCRIPTION**      The DISKQUOTA ADD command appends individual entries to a quota file on the specified disk. Note that the quota file must already exist and be enabled.

Unless you specify the permanent quota and overdraft values, the utility applies the default values from the UIC entry [0,0]. You adjust UIC [0,0] with the DISKQUOTA MODIFY command.



---

## **EXAMPLE**

```
① SYSMAN> SET ENVIRONMENT/NODE=(ZURICH,ASCONA)
%SYSMAN-I-ENV, Current command environment:
    Individual nodes: ZURICH,ASCONA
    Username ALEXIS   will be used on nonlocal nodes.
② SYSMAN> SET PROFILE /PRIVILEGE=SYSPRV
③ SYSMAN> DISKQUOTA ADD [MKT,MORSE] /DEVICE=WORK1 -
    SYSMAN> /PERMQUOTA=200 /OVERDRAFT=50
④ SYSMAN> DISKQUOTA ADD PAYROLL /DEVICE=WORK1 /PERMQUOTA=1000
```

- ① Defines the management environment to be nodes ZURICH and ASCONA.
- ② Adds SYSPRV privilege to the user's current privileges in order to write to the quota file.
- ③ Adds UIC [MKT,MORSE] to the quota file on the device named WORK1 on both nodes ZURICH and ASCONA, setting the permanent quota to 200 disk blocks and the overdraft limit to 50 disk blocks, for an absolute limit of 250 blocks.
- ④ Adds an entry for the rights identifier PAYROLL. Any user holding the PAYROLL identifier can use this disk space.

# SYSMAN

## DISKQUOTA CREATE

---

## DISKQUOTA CREATE

Creates and enables a quota file for a disk volume that does not currently contain one.

Requires write (W) access to the volume's master file directory (MFD), plus one of the following: SYSPRV privilege, a system UIC, or ownership of the volume.

---

### FORMAT                    DISKQUOTA CREATE

---

### PARAMETERS            *None.*

---

### QUALIFIERS            ***/DEVICE=device-spec***

Specifies the disk volume on which to create a quota file. SYSMAN validates the device specification. A logical name may be specified for **device-spec**. If so, it is translated in the target environment.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

---

### DESCRIPTION            The DISKQUOTA CREATE command creates a quota file for a volume that does not currently have one.

Only one quota file, [000000]QUOTA.SYS, can be present on any volume or volume set. As soon as you create a quota file, establish default values for quotas and overdrafts by adjusting UIC [0,0] with the DISKQUOTA MODIFY command. When a disk has existing files, use the DISKQUOTA REBUILD command to have SYSMAN update the quota file to contain current usage values.

**Note:** Digital recommends that you do not create and enable a quota file on the system disk.

---

**EXAMPLE**

```
SYSMAN> SHOW ENVIRONMENT
%SYSMAN-I-ENV, Current command environment:
      Node ATHENS of local cluster
      Username ALEXIS   will be used on nonlocal nodes

SYSMAN> DO SHOW DEVICES
.
.
.
SYSMAN> DISKQUOTA CREATE /DEVICE=DJA31:
SYSMAN> DISKQUOTA MODIFY /DEVICE=DJA31: [0,0] -
_SYSMAN> /PERMQUOTA=10000 /OVERDRAFT=100
```

The commands in this example sequence display the characteristics of the current management environment and verify the device name. Then they create a quota file on the disk DJA31 and set up default quota values.

**SYSMAN**  
**DISKQUOTA DELETE**

---

**DISKQUOTA DELETE**

Removes an entry from a quota file.

See the command DISKQUOTA REMOVE for more information. The DISKQUOTA REMOVE and DISKQUOTA DELETE commands perform the same function.

---

## **DISKQUOTA DISABLE**

Suspends the maintenance and enforcement of disk quotas on a volume.  
Requires SYSPRV privilege, a system UIC, or ownership of the volume.

---

**FORMAT**            **DISKQUOTA DISABLE**

---

**PARAMETERS**    *None.*

---

**QUALIFIERS**     ***/DEVICE=device-spec***  
Specifies a disk volume on which to disable a quota file. SYSMAN validates the device specification. A logical name may be specified for **device-spec**. If so, it is translated in the target environment.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

---

**DESCRIPTION**    The DISKQUOTA DISABLE command suspends quota operations on a volume. To permanently disable quotas on a device, disable the quotas with the DISKQUOTA DISABLE command and delete the file QUOTA.SYS. Otherwise, the system implicitly enables quotas when the disk is mounted, leaving invalid quota information.

If you enable the quota file later, enter the DISKQUOTA REBUILD command to update UIC entries and usage counts.

---

### **EXAMPLE**

```
SYSMAN> SET ENVIRONMENT/NODE=AMANDA  
SYSMAN> DISKQUOTA DISABLE /DEVICE=DJA1:
```

The command in this example suspends quota enforcement on disk DJA1, located on node AMANDA.

# SYSMAN

## DISKQUOTA ENABLE

---

## DISKQUOTA ENABLE

Resumes quota enforcement on a disk volume.

Requires SYSPRV privilege, a system UIC, or ownership of the volume.

---

### FORMAT            DISKQUOTA ENABLE

---

**PARAMETERS**    *None.*

---

### QUALIFIERS        */DEVICE=device-spec*

Specifies a disk volume on which to enable the quota file. SYSMAN validates the device specification. A logical name may be specified for **device-spec**. If so, it is translated in the target environment.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

---

**DESCRIPTION**    The DISKQUOTA ENABLE command reinstates the enforcement of quotas on a volume that had been suspended with the DISKQUOTA DISABLE command. Whenever you enable quotas on a volume, use the DISKQUOTA REBUILD command to update UIC entries and usage counts.

---

### EXAMPLE

```
SYSMAN> SET ENVIRONMENT/NODE=BAKER
SYSMAN> SET PROFILE/DEFAULT=DJA12:[ALEXIS.MGR]
SYSMAN> DISKQUOTA ENABLE
SYSMAN> DISKQUOTA REBUILD
```

The command in this example resumes quota enforcement on the default disk DJA12, which is located on node BAKER. The DISKQUOTA REBUILD command updates the quota file, correcting quotas and adding any new entries.

---

## DISKQUOTA MODIFY

Changes an entry in a quota file or adjusts default values for quotas and overdrafts. If a new quota limit is less than the current usage count, the utility issues a warning message before it implements the new quota.

Requires write (W) access to the quota file.

---

**FORMAT**            **DISKQUOTA MODIFY**    *uic*

---

**PARAMETER**        *uic*

Specifies the user identification code (UIC). You can specify the UIC in numeric or alphanumeric format. For complete information on UIC specification, refer to the *VMS DCL Dictionary*.

You can also specify quota entries for rights identifiers. These are rights granted a user with the AUTHORIZE Utility. Rights identifiers use an ID format rather than a UIC format. See the *VMS System Services Reference Manual* for a complete description.

When working in nonlocal environments, make sure that the alphanumeric UIC or rights identifiers that you use are valid for the environment.

---

**QUALIFIERS**        ***/DEVICE=device-spec***

Specifies the disk volume that contains the quota file. SYSMAN validates the device specification. A logical name may be specified for **device-spec**. If so, it is translated in the target environment.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

***/OVERDRAFT=value***

Specifies a positive integer that provides an overdraft value for the specified UIC. If you omit a value, the overdraft value defaults to the overdraft value in the entry for [0,0].

***/PERMQUOTA=value***

Specifies a positive integer that provides the quota for the specified UIC. If you omit a value, the permanent quota defaults to the value of the quota in the entry for [0,0].

---

**DESCRIPTION**

The DISKQUOTA MODIFY command changes values in a quota file for the disk named in the device specification. If you establish a quota limit that is less than the current usage count, a user can still log in and out, but cannot create files.

# **SYSMAN**

## **DISKQUOTA MODIFY**

After creating a quota file, use the DISKQUOTA MODIFY command to set default values for quotas and overdrafts. UIC [0,0] sets the default permanent quota and overdraft values for a quota file, so you must change the entry [0,0] to values appropriate for your installation. Unless you specify quota and overdraft values when adding a file entry, the utility applies these defaults to UIC entries.

---

### **EXAMPLES**

**1** SYSMAN> SET ENVIRONMENT/NODE=SIREN  
SYSMAN> DISKQUOTA MODIFY /DEVICE=DUA12: [0,0] -  
\_SYSMAN> /PERMQUOTA=3000 /OVERDRAFT=300

The command in this example edits the entry for UIC [0,0] in the quota file on DUA12, which is located on node SIREN.

**2** SYSMAN> DISKQUOTA MODIFY /DEVICE=SYS\$DISK1 [TTD,DAVIS] -  
\_SYSMAN> /PERMQUOTA=900

The command in this example sets the permanent quota for UIC [TTD,DAVIS] to 900 blocks, while making no change to the overdraft limit. SYSMAN modifies the quota file that is located on disk SYS\$DISK1 in the current environment.



---

## **DISKQUOTA REBUILD**

Updates a quota file, adding new UICs and correcting usage counts for each user on the volume.

Requires write (W) access to the quota file, plus one of the following: SYSPRV privilege, a system UIC, or ownership of the volume.

---

|               |                          |
|---------------|--------------------------|
| <b>FORMAT</b> | <b>DISKQUOTA REBUILD</b> |
|---------------|--------------------------|

---

|                   |              |
|-------------------|--------------|
| <b>PARAMETERS</b> | <i>None.</i> |
|-------------------|--------------|

---

|                   |  |
|-------------------|--|
| <b>QUALIFIERS</b> | <b><i>/DEVICE=device-spec</i></b><br>Specifies the disk volume that contains the quota file. SYSMAN validates the device specification and translates any logical name in the target environment before rebuilding the file.<br><br>Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment. |
|-------------------|--|

---

|                    |   |
|--------------------|---|
| <b>DESCRIPTION</b> | The DISKQUOTA REBUILD command reads the disk, and updates usage counts for all existing entries and adds new entries. It sets quota and overdraft values to the defaults set in UIC [0,0] if the entry did not previously exist. While the REBUILD command is executing, file activity on the volume is frozen. No files can be created, deleted, extended, or truncated. |
|--------------------|---|

Use the DISKQUOTA REBUILD command in the following circumstances:

- After creating a quota file on a volume with existing files.
- When the quota file has been enabled after a period of being disabled. The command corrects the usage counts and adds any new UICs.

---

## **EXAMPLE**

```
SYSMAN> SET ENVIRONMENT /NODE=WEST
SYSMAN> SET PROFILE /PRIVILEGE=SYSPRV
SYSMAN> DISKQUOTA ENABLE /DEVICE=DUA226:
SYSMAN> DISKQUOTA REBUILD /DEVICE=DUA226:
```

The command in this example enables the quota file and reconstructs the usage counts for all entries on disk DUA226, which is located on node WEST.

**SYSMAN**  
**DISKQUOTA REMOVE**

---

**DISKQUOTA REMOVE**

Removes an entry from a quota file.  
Requires write (W) access to the quota file.

---

**FORMAT**            **DISKQUOTA REMOVE**    *uic*

---

**PARAMETER**    *uic*

Specifies the user identification code (UIC). You can specify the UIC in numeric or alphanumeric format. For complete information on UIC specification, refer to the *VMS DCL Concepts Manual*.

You can also specify quota entries for rights identifiers. These are rights granted a user with the AUTHORIZE Utility. Rights identifiers use an ID format rather than a UIC format. See the *VMS System Services Reference Manual* for a complete description.

When working in nonlocal environments, be careful that the alphanumeric UIC or rights identifiers that you use are valid for the environment.

---

**QUALIFIERS**    */DEVICE=device-spec*

Specifies the disk volume containing the quota file. SYSMAN validates the device specification and translates any logical name in the target environment before deleting the UIC entry.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

---

**DESCRIPTION**    The DISKQUOTA REMOVE command eliminates the specified UIC from the quota file on the named device.

If the usage count for the UIC is not zero, the utility issues a warning message before it removes the UIC. Files remain on disk, and the user can still log on; however, any attempt to create files will fail.

The UIC [0,0] entry cannot be removed.

---

**EXAMPLE**

```
SYSMAN> SET ENVIRONMENT/NODE=MARS
SYSMAN> SHOW PROFILE
%SYSMAN-I-DEFDIR, Default directory on node MARS -- WORK2:[CASEY]
%SYSMAN-I-DEFPRIV, Process privileges on node MARS --
    TMPMGX
    OPER
    NETMBX
    SYSPRV
SYSMAN> DISKQUOTA REMOVE /DEVICE=DUA45: [TTD,DAVIS]
```

The command in this example removes UIC [TTD,DAVIS] from the quota file for disk DUA45, which is located on node MARS.

# SYSMAN

## DISKQUOTA SHOW

---

## DISKQUOTA SHOW

Displays quotas, overdrafts, and usage counts.

Requires no additional privileges to display your own quota, overdraft, and usage count, but otherwise requires read (R) access to the quota file.

---

### FORMAT            DISKQUOTA SHOW    *uic*

---

#### PARAMETER

##### ***uic***

Specifies the user identification code (UIC). You can specify the UIC in numeric or alphanumeric format. For complete information on UIC specification, refer to the *VMS DCL Concepts Manual*.

You can also specify quota entries for rights identifiers. These are rights granted a user with the AUTHORIZE Utility. Rights identifiers use an ID format rather than a UIC format. See the *VMS System Services Reference Manual* for a complete description.

You can use an asterisk wildcard character (\*) to specify the quota entry as follows:

| Command                 | Description                 |
|-------------------------|-----------------------------|
| DISKQUOTA SHOW [TTD,CJ] | Show user CJ in group TTD   |
| DISKQUOTA SHOW [TTD,*]  | Show all users in group TTD |
| DISKQUOTA SHOW *        | Show all entries            |

---

#### QUALIFIERS

##### ***/DEVICE=device-spec***

Specifies the disk volume containing the quota file. DISKQUOTA validates device specification and translates any logical name in the target environment before displaying UIC entries.

Without a device specification, SYSMAN uses the default disk on the target node. Unless you have set a default device with the SET PROFILE command, the default disk is the current device on the local node or the login default on another node, depending on the established environment.

##### ***/OUTPUT[=filespec]***

Directs output to the specified file. Without a file specification, /OUTPUT defaults to SYSMAN.LIS in the current directory on the local node where you are running SYSMAN.

---

**EXAMPLE**

SYSMAN> DISKQUOTA SHOW [ACCT,\*]

The command in this example displays quotas, overdrafts, and usage counts for all users in group ACCT on the default disk.

# SYSMAN

## DO

---

## DO

Executes a DCL command or DCL command procedure on all nodes in the current environment.

Requires the privileges of the DCL command being executed.

---

**FORMAT**            **DO** *[command-line]*

---

**PARAMETER**        ***command-line***  
Specifies a command string that SYSMAN passes to the DCL for execution. For complete information on DCL command syntax, refer to the *VMS DCL Dictionary*.

---

**QUALIFIERS**        ***/OUTPUT[=filespec]***  
Records output from the command in the specified file, which is located on the node from which you are executing SYSMAN. Position the qualifier immediately after the DO command. The default file specification is SYSMAN.LIS in the current device and directory. SYSMAN prefaces output with the message “%SYSMAN-I-OUTPUT, command execution on node xxxxxx.”

---

**DESCRIPTION**      The DO command executes the accompanying DCL command or DCL command procedure on all nodes in the current environment. Each DO command executes as an independent process, so there is no process context retained between DO commands. For this reason, you must express all DCL commands in a single command string, and you cannot run a program that expects input.

In a cluster environment, SYSMAN executes the commands sequentially on all nodes in the cluster. Each command executes completely before SYSMAN sends it to the next node in the environment. Any node that is unable to execute the command returns an error message. The utility displays an error message if the timeout period expires before the node responds. Some DCL commands, such as MOUNT/CLUSTER, operate clusterwide by design. For these commands to execute successfully in SYSMAN, define the environment to be a single node within the cluster.

Use the RSX command MCR to run programs located in SYS\$SYSTEM. The MCR command allows you to run a program and supply a command in a single command string.

---

## EXAMPLES

**1** SYSMAN> SET ENVIRONMENT/CLUSTER/NODE=NONAME  
SYSMAN> DO/OUTPUT SHOW DEVICE

The first command in this example defines the management environment to be the cluster where NONAME is a member. The second command executes a DCL command on each node in the cluster. Output goes to the file SYSMAN.LIS rather than to the terminal.

**2** SYSMAN> SET PROFILE /PRIVILEGES=(CMKRNL, SYSPRV) -  
\_SYSMAN> /DEFAULT=SYS\$SYSTEM  
SYSMAN> DO INSTALL ADD /OPEN/SHARED WRKDS:[MAIN]STATSHR  
SYSMAN> DO MCR AUTHORIZE ADD JONES/PASSWORD=COLUMBINE -  
\_SYSMAN> /DEVICE=WORK1 /DIRECTORY=[JONES]

The first command in this example adds CMKRNL and SYSPRV privileges to the current privileges because they are required by the INSTALL and the AUTHORIZE utility. The next command installs the file STATSHR. The last command sets up an account for user JONES, specifying a password as well as a default device and directory.

The MCR command in the last line of the example allows you to invoke the Authorize Utility from SYS\$SYSTEM and add a record to the UAF in one command string.

**3** SYSMAN> SET ENVIRONMENT/NODE=LONDON  
SYSMAN> SET PROFILE /DEFAULT=[CJ.PROGRAMS] -  
\_SYSMAN> /PRIVILEGES=NOSYSRV  
SYSMAN> DO/OUTPUT @PROCESS\_INFO

The commands in this example define the environment to be a single node and adjust the current privileges and directory. The DO command executes the command procedure PROCESS\_INFO.COM, located in directory [CJ.PROGRAMS] and writes any output to SYSMAN.LIS in the directory from which SYSMAN is running.

**SYSMAN**  
**EXIT**

---

**EXIT**

Terminates the SYSMAN session and returns control to the DCL command level. Any profile changes, established on the local node with the command SET PROFILE, are restored to their values at the time SYSMAN was invoked. You can also press Ctrl/Z to exit at any time.

---

**FORMAT**            **EXIT**

---

**PARAMETERS**    *None.*

---

**QUALIFIERS**     *None.*



---

## **HELP**

Provides online help information for using the SYSMAN commands, parameters, and qualifiers. Press Ctrl/Z to exit.

---

**FORMAT**            **HELP** *[keyword...]*

---

**PARAMETER**        *keyword*  
Specifies the command, parameter, or qualifier for which help information is to be displayed. If you omit the keyword, the HELP command displays a list of available help topics and prompts you for a particular keyword.

---

**QUALIFIERS**        *None.*

---

## **EXAMPLE**

SYSMAN> HELP DO

The command in this example displays help information about the SYSMAN command, DO.

# SYSMAN

## LICENSE LOAD

---

## LICENSE LOAD

Activates licenses registered in the LICENSE database.  
Requires CMKRNL, SYSNAM, and SYSPRV privileges.

---

**FORMAT**            **LICENSE LOAD** *product*

---

**PARAMETER**        *product*  
Specifies the name of the product whose license you want to activate.

---

**QUALIFIERS**        ***/DATABASE=filespec***  
Allows you to specify the location of the LICENSE database. The default file specification is SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB. You do not need to use the /DATABASE qualifier if you use the default LICENSE database name and location.

***/PRODUCER=string***  
Allows you to specify the name of the company that owns the product for which you have a license. Use this qualifier only if the product is from a company other than Digital.

---

**DESCRIPTION**      You can use the LICENSE LOAD command to activate licenses on multiple systems and on nonlocal systems in the system management environment. The SYSMAN LICENSE commands are a subset of the License Management Facility (LMF) commands. For more information about the LMF, see the *VMS License Management Utility Manual*.

---

## EXAMPLE

SYSMAN> LICENSE LOAD FORTRAN

The command in this example activates the license for VAX FORTRAN. Because the license is for a Digital product, the command does not include the /PRODUCER qualifier.

---

## **LICENSE UNLOAD**

Deactivates licenses registered in the LICENSE database.

Requires CMKRNL, SYSNAM, and SYSPRV privileges.

---

**FORMAT**            **LICENSE UNLOAD** *[product]*

---

**PARAMETER**    *product*  
Specifies the name of the product whose license you want to deactivate. If you enter the LICENSE UNLOAD command without specifying a product name, the system deactivates all available registered licenses.

---

**QUALIFIERS**    */PRODUCER=string*  
Allows you to specify the name of the company that owns the product for which you have a license. Use this qualifier only if the product is from a company other than Digital.

---

**DESCRIPTION**    The LICENSE UNLOAD command can be used to deactivate licenses on multiple systems and on nonlocal systems in the system management environment. The SYSMAN LICENSE commands are a subset of the License Management Facility (LMF) commands. For more information about the LMF, see the *VMS License Management Utility Manual*.

---

### **EXAMPLE**

SYSMAN> LICENSE UNLOAD FORTRAN

This command deactivates the license for VAX FORTRAN. Because the license is for a Digital product, the command does not include the /PRODUCER qualifier.

# SYSMAN

## PARAMETERS DISABLE CHECKS

---

## PARAMETERS DISABLE CHECKS

Bypasses validation of parameter values. SYSMAN parameter validation consists of ensuring that the parameters fall within the defined minimum and maximum values specified in the PARAMETERS SET command.

---

### FORMAT                      PARAMETERS DISABLE CHECKS

---

**PARAMETERS**     *None.*

---

**QUALIFIERS**        *None.*

---

**DESCRIPTION**     The PARAMETERS DISABLE CHECKS command allows you to override minimum and maximum values established for system parameters. Parameter checks are *enabled* by default. If you attempt to set parameter values outside the allowable limits when checks are enabled, the operating system issues an error message. By disabling checks you can set parameter values regardless of the minimum and maximum limits.

**Note:** Range checks are enabled by default because Digital suggests that systems operate within these minimum and maximum values. Setting parameters outside these limits can result in system failures or hangs.

---

### EXAMPLE

```
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> SET PROFILE/DEFAULT=SYS$SYSTEM/PRIVILEGES=CMEEXEC
SYSMAN> PARAMETERS SET MAXPROCESSCNT 10
%SMI-E-OUTRANGE, parameter is out of range
SYSMAN> PARAMETERS DISABLE CHECKS
SYSMAN> PARAMETERS SET MAXPROCESSCNT 10
```

In this example, the initial attempt to set MAXPROCESSCNT below the minimum fails because range checks are enabled. However, once range checks are disabled, the PARAMETERS SET MAXPROCESSCNT command succeeds.

---

## PARAMETERS ENABLE CHECKS

Validates all parameter values to ensure they fall within the defined minimum and maximum values.

Because range checks are enabled by default, use **PARAMETERS ENABLE CHECKS** after entering a **PARAMETERS DISABLE CHECKS** command.

---

### FORMAT                    **PARAMETERS ENABLE CHECKS**

---

**PARAMETERS**     *None*

---

**QUALIFIERS**        *None.*

---

### EXAMPLE

```
SYSMAN> PARAMETERS DISABLE CHECKS
SYSMAN> PARAMETERS SET WSMAX 20
SYSMAN> PARAMETERS ENABLE CHECKS
SYSMAN> PARAMETERS SET WSMAX 30
%SMI-E-OUTRANGE, parameter is out of range
SYSMAN> PARAMETERS SHOW WSMAX
Parameter Name    Current    Default    Minimum    Maximum    Unit    Dynamic
WSMAX            2000     1024       60       6400    pages
```

The **PARAMETERS ENABLE CHECKS** command in this example illustrates that when range checking is disabled, the system accepts a working set value (WSMAX) of 20. However, once range checking is enabled with the **PARAMETERS ENABLE CHECKS** command, the system does not accept a WSMAX below the minimum, which is 60.

# SYSMAN

## PARAMETERS SET

---

## PARAMETERS SET

Changes the value of a specific parameter in the work area.

The PARAMETERS SET command does not modify parameter files, the current system parameter file on disk, or the active system. For information on performing these modifications, see the PARAMETERS WRITE command.

---

|               |   |
|---------------|---|
| <b>FORMAT</b> | <b>PARAMETERS SET</b> <i>parameter-name</i><br><i>value</i><br><i>/STARTUP filespec</i> |
|---------------|---|

---

|                   |   |
|-------------------|---|
| <b>PARAMETERS</b> | <b><i>parameter-name</i></b><br>Specifies the name of the parameter to modify. Instead of a name, you can enter a period (.) to change the value of the most recently displayed or the most recently modified parameter. See the PARAMETERS SHOW command for an example of using the period in place of a parameter name. |
|-------------------|---|

For a list of system parameters and further information on them, use the command HELP PARAMETERS.

### ***value***

Specifies the new value for the parameter. Enclose values for ASCII parameters in quotation marks if they contain embedded spaces or other special characters.

Typically the value is an integer or the keyword DEFAULT. The keyword DEFAULT sets the parameter to its default value. The PARAMETERS SHOW command displays the defined minimum, maximum, and default values for the parameter, which are required unless range checking is disabled with the command PARAMETERS DISABLE CHECKS.

---

|                   |   |
|-------------------|---|
| <b>QUALIFIERS</b> | <b><i>/STARTUP filespec</i></b><br>Sets the name of the site-independent startup procedure to the given file specification. A file specification has a maximum length of 31 characters. The initial startup command procedure is SYS\$SYSTEM:STARTUP.COM. |
|-------------------|---|

---

## EXAMPLES

**1** SYSMAN> PARAMETERS SET PFCDEFAULT 20

The PARAMETERS SET command in this example assigns a value of 20 to the PFCDEFAULT parameter.

# **SYSMAN PARAMETERS SET**

**2** SYSMAN> PARAMETERS SET GBLSECTIONS DEFAULT

The **PARAMETERS SET** command in this example assigns the default value (40) to the **GBLSECTIONS** parameter.

**3** SYSMAN> PARAMETERS SET/STARTUP SYS\$SYSTEM:XSTARTUP.COM

The command in this example assigns **SYS\$SYSTEM:XSTARTUP.COM** as the current site-independent startup command procedure.

# SYSMAN

## PARAMETERS SHOW

---

## PARAMETERS SHOW

Displays the value of a parameter or a group of parameters in the work area. In addition, the command shows the minimum, maximum, and default values of a parameter and its unit of measure.

---

**FORMAT**            **PARAMETERS SHOW**    *[parameter-name]*

---

**PARAMETER**        ***parameter-name***  
Specifies the name of a parameter or a period (.). A period is interpreted as a request for the parameter specified in the last PARAMETERS SET or PARAMETERS SHOW command. The parameter name can be abbreviated, but the abbreviation must be unique because SYSMAN selects the first parameter that matches.

---

**QUALIFIERS**        ***/ACP***  
Displays all Files-11 ACP parameters.

***/ALL***  
Displays the values of all active parameters.

***/CLUSTER***  
Displays all parameters specific to clusters.

***/DYNAMIC***  
Displays all parameters that would be in effect immediately after you enter a PARAMETERS WRITE ACTIVE command.

***/GEN***  
Displays all general parameters.

***/HEX***  
Displays numeric parameters in hexadecimal rather than decimal radix. Specify the /HEX system parameter name or the parameter type. If you specify the /HEX qualifier with the /NAMES qualifier, /HEX is ignored.

***/JOB***  
Displays all Job Controller parameters.

***/LGI***  
Displays all LOGIN security control parameters.

***/MAJOR***  
Displays the most important parameters.

***/MULTIPROCESSING***  
Displays parameters specific to multiprocessing.



# **SYSMAN PARAMETERS SHOW**

## ***/NAMES***

Displays only parameter names. You can combine other qualifiers with this one.

## ***/OUTPUT***

Directs output to the specified file rather than SYS\$OUTPUT. Without a file specification, the output goes to SYSMAN.LIS in the current directory.

## ***/PQL***

Displays the parameters for all default process quotas.

## ***/RMS***

Displays all parameters specific to VMS Record Management Services (VMS RMS).

## ***/SCS***

Displays all parameters specific to VAXcluster System Communication Services.

## ***/SPECIAL***

Displays all special control parameters.

## ***/STARTUP***

Displays the name of the site-independent startup procedure.

## ***/SYS***

Displays all active system parameters.

## ***/TTY***

Displays all parameters for terminal drivers.

---

## **DESCRIPTION**

Parameters are displayed in decimal unless the */HEX* qualifier is specified. Note that ASCII values are always displayed in ASCII.

Abbreviations for parameter names must be unique because the first parameter matching the abbreviation is selected for display. No ambiguity checks are made. For example, a specification of *PARAMETERS SHOW GBL* displays the *GBLSECTIONS* parameter. To display the *GBLPAGFIL* parameter, you must specify *PARAMETERS SHOW GBLPAGF* to avoid further ambiguity with the *GBLPAGES* parameter.

You can use a period (.) to indicate that you want to work with the system parameter that was specified in the last *PARAMETERS SET* or *PARAMETERS SHOW* command.

# SYSMAN

## PARAMETERS SHOW

### EXAMPLES

```

1  SYSMAN> PARAMETERS SHOW GBLSECTIONS
Parameter Name      Current   Default   Minimum   Maximum Unit   Dynamic
GBLSECTIONS         100      40        20        -1 Sections

SYSMAN> PARAMETERS SET . 110
SYSMAN> PARAMETERS SHOW .
Parameter Name      Current   Default   Minimum   Maximum Unit   Dynamic
GBLSECTIONS         110      40        20        -1 Sections
  
```

In this example, the user first displays the values of the GBLSECTIONS parameter and then refers to the parameter with a period to set its current value to 110. The next PARAMETERS SHOW command also uses the period notation to obtain confirmation that the change occurred.

```

2  SYSMAN> PARAMETERS SHOW/ACP
  
```

The PARAMETERS SHOW command in this example produces output similar to the following:

```

Parameters in use: Active
Parameter Name      Current   Default   Minimum   Maximum Unit   Dynamic
ACP_MULTIPLE        0         1         0         1 Boolean   D
ACP_SHARE            1         1         0         1 Boolean
ACP_MAPCACHE        52        8         1         -1 Pages   D
ACP_HDRCACHE        138       128       2         -1 Pages   D
ACP_DIRCACHE        138       80        2         -1 Pages   D
ACP_DINDXCACHE      37        25        2         -1 Pages   D
ACP_WORKSET         0         0         0         -1 Pages   D
ACP_FIDCACHE        64        64        0         -1 File-Ids D
ACP_EXTCACHE        64        64        0         -1 Extents  D
ACP_EXTLIMIT        300       300       0         1000 Percent/10 D
ACP_QUOCACHE        130       64        0         -1 Users   D
ACP_SYSACC          4         8         0         -1 Directories D
ACP_MAXREAD         32        32        1         64 Blocks  D
ACP_WINDOW          7         7         1         -1 Pointers D
ACP_WRITEBACK       1         1         0         1 Boolean   D
ACP_DATACHECK       2         2         0         3 Bit-mask  D
ACP_BASEPRIO        8         8         4         31 Priority  D
ACP_SWAPFLGS        14        15        0         15 Bit-mask  D
ACP_XQP_RES         1         1         0         1 Boolean   D
ACP_REBLDSYS        0         1         0         1 Boolean
  
```

# SYSMAN PARAMETERS SHOW

3 SYSMAN> PARAMETERS SHOW/ACP/HEX

The PARAMETERS SHOW command in this example produces a hexadecimal display of the values of the ACP system parameters.

```
Parameters in use: Active
Parameter Name      Current   Default   Minimum   Maximum   Unit   Dynamic
ACP_MULTIPLE        00000000 00000001 00000000 00000001 Boolean  D
ACP_SHARE            00000001 00000001 00000000 00000001 Boolean
ACP_MAPCACHE         00000034 00000008 00000001 FFFFFFFF Pages    D
ACP_HDRCACHE         0000008A 00000080 00000002 FFFFFFFF Pages    D
ACP_DIRCACHE         0000008A 00000050 00000002 FFFFFFFF Pages    D
ACP_DNDXCACHE        00000025 00000019 00000002 FFFFFFFF Pages    D
ACP_WORKSET          00000000 00000000 00000000 FFFFFFFF Pages    D
ACP_FIDCACHE         00000040 00000040 00000000 FFFFFFFF File-Ids  D
ACP_EXTCACHE         00000040 00000040 00000000 FFFFFFFF Extents  D
ACP_EXTLIMIT         0000012C 0000012C 00000000 000003E8 Percent/10 D
ACP_QUOCACHE         00000082 00000040 00000000 FFFFFFFF Users    D
ACP_SYSACC           00000004 00000008 00000000 FFFFFFFF Directories D
ACP_MAXREAD          00000020 00000020 00000001 00000040 Blocks   D
ACP_WINDOW           00000007 00000007 00000001 FFFFFFFF Pointers  D
ACP_WRITEBACK        00000001 00000001 00000000 00000001 Boolean  D
ACP_DATACHECK        00000002 00000002 00000000 00000003 Bit-mask  D
ACP_BASEPRIO         00000008 00000008 00000004 0000001F Priority  D
ACP_SWAPFLGS         0000000E 0000000F 00000000 0000000F Bit-mask  D
ACP_XQP_RES          00000001 00000001 00000000 00000001 Boolean
ACP_REBLDSYS         00000000 00000001 00000000 00000001 Boolean
```

4 SYSMAN> PARAMETERS SHOW/STARTUP  
Startup command file = SYS\$SYSTEM:STARTUP.COM

The PARAMETERS SHOW command in this example displays the name of the site-independent startup command procedure.

# SYSMAN

## PARAMETERS USE

---

## PARAMETERS USE

Reads a set of system parameters into the work area for inspection or manipulation.

---

|               |                                     |
|---------------|-------------------------------------|
| <b>FORMAT</b> | <b>PARAMETERS USE</b> <i>source</i> |
|---------------|-------------------------------------|

---

### PARAMETER

#### ***source***

The source of a system parameter file for data to be read into the work area. The source can be any of the following:

- **ACTIVE**—Read parameters from the currently running system. When the work area is empty, this is the default.
- **CURRENT**—Read parameters from the disk image of the currently running system. Use of the **CURRENT** parameter requires read (R) access to SYS\$SYSTEM:VAXVMSSYS.PAR.
- **DEFAULT**—Read a parameter set containing the default values for all parameters.
- **filespec**—Read parameters from a previously created system parameter file. The default file type is PAR. Use of the parameter requires read access to the file.

---

### QUALIFIERS

*None.*

---

### DESCRIPTION

The **PARAMETERS USE** command initializes the work area to use the values of a new parameter file, the current system parameter file, or the default values, if the active values do not provide a suitable base for subsequent operations.

---

### EXAMPLE

```
SYSMAN> PARAMETERS USE DEFAULT
```

The **PARAMETERS USE** command in this example initializes the work area with parameter values that should allow the VMS operating system to boot on any standard configuration. The initial values of the work area when the utility is invoked are the active values.

---

## PARAMETERS WRITE

Writes the contents of the work area to the specified destination.

---

**FORMAT**                    **PARAMETERS WRITE**    *destination*

---

**PARAMETER**            *destination*

The destination of a new parameter file can be any of the following:

- **ACTIVE**—Write the parameter set to the currently running system. Use of the **ACTIVE** parameter requires the **CMKRNL** privilege.
- **CURRENT**—Write the parameter set to the disk image of the currently running system. The disk image is the current system parameter file on disk. Use of the **CURRENT** parameter requires write access to **SYS\$SYSTEM:VAXVMSSYS.PAR**.
- **filespec**—Create the given file and write the parameter set to it. The default file type is **PAR**. Use of this parameter requires write access to the file.

---

**QUALIFIERS**            *None.*

---

**DESCRIPTION**        The **PARAMETERS WRITE** command writes the system parameter values and the name of the site-independent startup command procedure from the work area to your choice of a parameter file, the current system parameter file on disk, or the active system in memory. (Only the dynamic parameter values are written to the active system.)

Both the **PARAMETERS WRITE ACTIVE** and **PARAMETERS WRITE CURRENT** commands send a message to **OPCOM** to record the event.

---

### EXAMPLES

**1**    **SYSMAN> PARAMETERS WRITE SYS\$SYSTEM:SPECIAL**

The command in this example creates a new parameter specification file.

**2**    **SYSMAN> PARAMETERS WRITE CURRENT**

The command in this example modifies the current system parameter file on disk (**SYS\$SYSTEM:VAXVMSSYS.PAR**).

# SYSMAN

## SET ENVIRONMENT

---

## SET ENVIRONMENT

Defines the nodes or cluster to which subsequent commands apply.  
Requires OPER or SETPRV privilege on all nodes in the target environment.

---

|               |                        |
|---------------|------------------------|
| <b>FORMAT</b> | <b>SET ENVIRONMENT</b> |
|---------------|------------------------|

---

|                   |              |
|-------------------|--------------|
| <b>PARAMETERS</b> | <i>None.</i> |
|-------------------|--------------|

---

### QUALIFIERS

#### ***/CLUSTER***

Directs SYSMAN to apply subsequent commands to all nodes in the cluster. By default, the management environment is the local cluster. Specify a nonlocal cluster by naming one cluster member with the */NODE* qualifier.

#### ***/NODE=(node1,node2,...)***

Specifies that SYSMAN execute subsequent commands on the given nodes. If accompanied by the */CLUSTER* qualifier, the environment becomes the cluster where the given node is a member. A node name can be a system name, cluster alias, or logical name. However, before you can use logical names to define the command environment, you must set up the logical name table SYSMAN\$NODE\_TABLE. For more information, see Section 1.1.

#### ***/USERNAME=username***

Specifies that this user name should be used for access control purposes on another node. SYSMAN uses the current user name if none is supplied. The utility prompts for a password whenever a new user name is specified.

---

### DESCRIPTION

The SET ENVIRONMENT command defines the target nodes or cluster for subsequent commands. When invoked, the system management environment is the local node where you are running SYSMAN. You can change the environment to any other nodes in the cluster, the entire cluster, or any nodes or cluster available through DECnet.

Designate a cluster environment with the */CLUSTER* qualifier. When specifying a nonlocal cluster, also include the */NODE* qualifier to identify the cluster.

You can display the current environment with the command SHOW ENVIRONMENT. To adjust privileges and defaults for the current environment, use the SET PROFILE command.

An environment exists until you exit from SYSMAN or establish another command context with the SET ENVIRONMENT command.

---

## EXAMPLES

**1** SYSMAN> SET ENVIRONMENT/CLUSTER  
%SYSMAN-I-ENV, Current command environment:  
Clusterwide on local cluster  
Username ALEXIS will be used on nonlocal nodes

The commands in this example define the command environment as the local cluster. SYSMAN confirms the new environment.

**2** SYSMAN> SET ENVIRONMENT/NODE=CLACK/CLUSTER  
Remote Password:  
%SYSMAN-I-ENV, Current command environment:  
Clusterwide on remote node CLACK  
Username ALEXIS will be used on nonlocal nodes

The command in this example establishes a management environment on the cluster where node CLACK is a member. SYSMAN prompts for a password because it is a nonlocal environment.

**3** SYSMAN> SET ENVIRONMENT/NODE=(LESETH, JOSHUA, TORIN)  
%SYSMAN-I-ENV, Current command environment:  
Individual nodes: LESETH, JOSHUA, TORIN  
Username ALEXIS will be used on nonlocal nodes

The command in this example defines the management environment to be three individual nodes.

**4** \$ CREATE/NAME\_TABLE/PARENT=LNMSYSTEM\_DIRECTORY -  
\_ \$ SYSMAN\$NODE\_TABLE  
\$ DEFINE LAVCS SYS1, SYS2, SYS3, SYS4/TABLE=SYSMAN\$NODE\_TABLE  
\$ RUN SYS\$SYSTEM:SYSMAN  
SYSMAN> SET ENVIRONMENT/NODE=(LAVCS)  
%SYSMAN-I-ENV, Current command environment:  
Individual nodes: SYS1, SYS2, SYS3, SYS4  
Username ALEXIS will be used on nonlocal nodes

The commands in this example set up the logical name table SYSMAN\$NODE\_TABLE, define a logical name (LAVCS), and use the logical name to define the command environment.

# SYSMAN

## SET PROFILE

---

## SET PROFILE

Temporarily modifies a user's current privileges and default device and directory.

---

### FORMAT            SET PROFILE

---

**PARAMETERS**    *None.*

---

**QUALIFIERS**    ***/DEFAULT=device:[directory]***  
Specifies the default disk device and directory name that the system should use in this environment to locate and catalog files.

***/PRIVILEGES=(priv1,priv2...)***  
Specifies the privileges to add to the current privileges. Any enhanced privileges must be authorized.

***/VERIFY***  
***/NOVERIFY (default)***  
Allows you to specify whether you want DCL verification (both procedure and image) for future DO commands.

---

**DESCRIPTION**    You need to consider the privilege requirements of commands that you will enter in an environment. The SET PROFILE command modifies process attributes for the current management environment. SYSMAN can add or delete current privileges, if they are authorized. It can also set a new default device and directory. You can also use the SET PROFILE/[NO]VERIFY command to control DCL command verification in SYSYMAN. Other attributes of your process remain constant. The profile is in effect until you change it, reset the environment, or exit from the utility. Section 3 discusses profile changes in more detail.

---

### EXAMPLES

**1**    SYSMAN> SET PROFILE/DEFAULT=WORK1:[ALEXIS]

The command in this example changes the default device and directory in the user account to directory ALEXIS on device WORK1.

**2**    SYSMAN> SET PROFILE/PRIVILEGES=(SYSPRV,CMKRNL)/VERIFY

The command in this example makes the authorized privileges, SYSPRV and CMKRNL, part of the current privileges, and turns on DCL verification. The privileges remain in effect until the environment changes, you enter another SET PROFILE command, or you exit.



---

## **SET TIMEOUT**

Establishes the amount of time SYSMAN waits for a node to respond. Once the time limit expires, SYSMAN proceeds to execute the command on the next node in the environment.

---

**FORMAT**            **SET TIMEOUT**    *time*

---

**PARAMETER**        *time*  
Specifies a delta time value, which has the following format:

hh:mm:ss[.cc.]

This is the amount of time that SYSMAN waits for a node to respond. By default, there is no timeout period, so SYSMAN waits indefinitely. See the *VMS DCL Concepts Manual* for a description of delta time values.

---

**QUALIFIERS**        *None.*

---

### **EXAMPLE**

```
SYSMAN> SET TIMEOUT 00:00:30
%SYSMAN-I-TIMEVAL, timeout value is 00:00:30
SYSMAN> CONFIGURATION SHOW TIME
System time on node ASCONA: 19-JUN-1990 14:22:33
%SYSMAN-I-NODERR, error returned from node LUGANO
%SMI-E-TIMEOUT, remote operation has timed out
System time on node JOSHUA: 19-JUN-1990 14:23:15
```

The command in this example establishes a timeout period of 30 seconds. Because node LUGANO did not respond within 30 seconds, SYSMAN displays an error message and proceeds to execute the command on the next node in the environment.

# SYSMAN

## SHOW ENVIRONMENT

---

# SHOW ENVIRONMENT

Displays the target nodes or cluster where SYSMAN is executing commands.

---

## FORMAT SHOW ENVIRONMENT

---

**PARAMETERS** *None.*

---

**QUALIFIERS** *None.*

---

**DESCRIPTION** The SHOW ENVIRONMENT command displays the current management environment. It can be the local cluster, local or remote nodes, or a nonlocal cluster. SYSMAN indicates if the environment is limited to individual nodes or if it is clusterwide. It also shows the current user name.

The environment exists until you exit from SYSMAN or enter another SET ENVIRONMENT command.

---

## EXAMPLES

**1** SYSMAN> SHOW ENVIRONMENT  
%SYSMAN-I-ENV, Current command environment:  
Clusterwide on local cluster  
Username ALEXIS will be used on nonlocal nodes

The command in this example shows the current environment is the local cluster. User name ALEXIS will be used on other nodes in the cluster.

**2** SYSMAN> SHOW ENVIRONMENT  
%SYSMAN-I-ENV, Current command environment:  
Clusterwide on remote cluster CLACK  
Username ALEXIS will be used on nonlocal nodes

The command in this example shows that the command environment is a nonlocal cluster where node CLACK is a member.

**3** SYSMAN> SHOW ENVIRONMENT  
%SYSMAN-I-ENV, Current command environment:  
Individual nodes: TURIN, JOSHUA  
At least one node is not in local cluster  
Username ALEXIS will be used on nonlocal nodes

The command in this example shows that the command environment consists of 2 nodes.

---

## **SHOW KEY**

Displays key definitions created with the DEFINE/KEY command.

---

**FORMAT**            **SHOW KEY** *[key-name]*

---

**PARAMETER**    **key-name**  
Specifies the name of the key whose definition you want displayed. See the DEFINE/KEY command for a list of valid key names.

---

**QUALIFIERS**    **/ALL**  
Displays all the key definitions in the specified state or states. You do not need to specify a key name.

**/BRIEF**  
Displays only the key definition. By default, the system displays all the qualifiers associated with the key definition, including any specified state, unless you use the /BRIEF qualifier.

**/DIRECTORY**  
Displays the names of all the states for which you have defined keys. If you have not defined keys, the SHOW KEY/DIRECTORY command displays the DEFAULT and GOLD states (which is the default SYSMAN keypad).

**/STATE=(state, state...)**  
Specifies the name of a state for which the specified key definitions are to be displayed. If you select more than one state name, separate them with commas and enclose the list in parentheses.

---

**DESCRIPTION**    Specifies the name of the key whose definition you want displayed. See the DEFINE/KEY command for a list of valid key names.

---

### **EXAMPLE**

```
SYSMAN> SHOW KEY/ALL  
  
DEFAULT keypad definitions:  
  KP0 = "SHOW ENVIRONMENT" (echo)  
  KP1 = "SHOW PROFILE" (echo)  
SYSMAN>
```

This SHOW KEY command in this example displays all of the key definitions currently in effect.

# **SYSMAN**

## **SHOW PROFILE**

---

# **SHOW PROFILE**

Displays the current privileges and the default device and directory being used in the current environment.

---

## **FORMAT**            **SHOW PROFILE**

---

**PARAMETERS**    *None.*

---

**QUALIFIERS**    ***/DEFAULT***  
Displays the default disk device and directory name that the system uses in this environment to locate and catalog files.

***/PRIVILEGES***  
Displays only the privileges currently in effect for the current environment.

---

**DESCRIPTION**    The **SHOW PROFILE** command displays the current privileges and the default device and directory that is being used in the current environment. You can modify these attributes with the **SET PROFILE** command.

These values remain in effect until you change environments or enter another **SET PROFILE** command.

---

## **EXAMPLE**

```
SYSMAN> SHOW PROFILE
%SYSMAN-I-DEFDIR, Default directory on node BAKER -- WORK1:[BERGERON]
%SYSMAN-I-DEFPRIV, Process privileges on node BAKER --
    TMPMGX
    OPER
    NETMBX
    SYSPRV
```

The command in this example shows the default device and directory as well as current privileges.

---

## **SHOW TIMEOUT**

Displays the amount of time SYSMAN waits for a node to respond. By default, there is no timeout period.

---

|               |                     |
|---------------|---------------------|
| <b>FORMAT</b> | <b>SHOW TIMEOUT</b> |
|---------------|---------------------|

---

|                  |              |
|------------------|--------------|
| <b>PARAMETER</b> | <i>None.</i> |
|------------------|--------------|

---

|                   |              |
|-------------------|--------------|
| <b>QUALIFIERS</b> | <i>None.</i> |
|-------------------|--------------|

---

### **EXAMPLE**

```
SYSMAN> SHOW TIMEOUT
%SYSMAN-I-TIMEVAL, timeout value is 00:00:04.00
```

The **SHOW TIMEOUT** command in this example displays the current timeout value, which is 4 seconds.

# SYSMAN

## SPAWN

---

## SPAWN

Creates a subprocess of the current process. The context of the subprocess is copied from the current process. You can use the SPAWN command to leave SYSMAN temporarily, perform other tasks (such as displaying a directory listing or printing a file), and return to SYSMAN.

Note that SPAWN performs actions on the local node only. If you want to execute DCL commands or command procedures throughout your environment, use the DO command.

Requires TMPMBX or PRMMBX user privilege. The SPAWN command does not manage terminal characteristics. The SPAWN and ATTACH commands cannot be used if your terminal has an associated mailbox.

---

**FORMAT**            **SPAWN** *[command-string]*

---

**PARAMETER**        ***command-string***  
Specifies a command string of fewer than 132 characters that you want executed in the context of the created subprocess. When the command completes execution, the subprocess terminates and control returns to the parent process. If both a command string and the /INPUT qualifier are specified, the specified command string executes before additional commands are obtained from the /INPUT qualifier.

---

**QUALIFIERS**        ***/INPUT=filespec***  
Specifies an input file containing one or more DCL command strings that you want executed by the spawned subprocess. If you specify a command string along with an input file, the command string gets processed before the commands in the input file. When processing is complete, the subprocess terminates.

***/LOGICAL\_NAMES (default)***  
***/NOLOGICAL\_NAMES***  
Specifies that the logical names of the parent process are copied to the subprocess. When you do not want the subprocess to use the logical names of the parent process, enter the /NOLOGICAL\_NAMES qualifier.

***/OUTPUT=filespec***  
Identifies the output file to which the results of the operation are written. You should specify an output other than SYS\$OUTPUT whenever you use the /NOWAIT qualifier. This prevents output from being displayed while you are specifying new commands. If you omit the /OUTPUT qualifier, output gets written to the current SYS\$OUTPUT device.

***/PROCESS=subprocess-name***  
Specifies the name of the subprocess that you want to create. The default subprocess name is in the format USERNAME\_n.

***/SYMBOLS (default)***

***/NOSYMBOLS***

Determines whether the system passes DCL global and local symbols to the subprocess.

***/WAIT***

***/NOWAIT***

Controls whether the system waits until the subprocess completes before allowing you to specify more commands. The */NOWAIT* qualifier allows you to specify new commands while the specified subprocess is running. If you specify the */NOWAIT* qualifier, you should also use the */OUTPUT* qualifier to direct the output to a file rather than displaying it on the screen. Doing this prevents your terminal from being used by more than one process simultaneously.

---

**DESCRIPTION**

The SPAWN command creates a subprocess of your current process with the following attributes copied from the parent process:

- All symbols except \$RESTART, \$SEVERITY, and \$STATUS
- Key definitions
- The current keypad state
- The current prompt string
- All process logical names and logical name tables except those explicitly marked CONFINE or those created in executive or kernel mode
- Default disk and directory
- Current SET MESSAGE settings
- Current process privileges
- Control and verification states

Note that some attributes, such as the process's current command tables, are *not* copied.

When the subprocess is created, the process-permanent open files and any image or procedure context are *not* copied from the parent process. The subprocess is set to command level 0 (DCL level with the current prompt).

If you do not specify the */PROCESS* qualifier, the name of this subprocess is composed of the same base name as the parent process and a unique number. For example, if the parent process name is SMITH, the subprocess name can be SMITH\_1, SMITH\_2, and so on.

The LOGIN.COM file of the parent process is not executed for the subprocess because the context is copied separately, allowing quicker initialization of the subprocess. When the */WAIT* qualifier is in effect, the parent process remains in hibernation until the subprocess terminates or returns control to the parent by way of the ATTACH command.

# SYSMAN

## SPAWN

More than one process simultaneously attempts to use the same input or output stream when several processes share that stream and you do one of the following:

- Terminate a subprocess to which you are not currently attached
- Terminate a process that is not spawned from the process to which you are currently attached

You should use the LOGOUT command to terminate the subprocess and return to the parent process. You can also use the ATTACH command (see ATTACH) to transfer control of the terminal to another process in the subprocess tree, including the parent process. (The SHOW PROCESS /SUBPROCESSES command displays the processes in the subprocess tree and points to the current process.)

**Note:** Because a tree of subprocesses can be established using the SPAWN command, you must be careful when terminating any process in the tree. When a process is terminated, all subprocesses below that point in the tree are automatically terminated.

Qualifiers used with the SPAWN command must directly follow the command verb. The command string parameter begins after the last qualifier and continues to the end of the command line.

---

## EXAMPLES

```
1  SYSMAN> SPAWN DIR SYSTARTUP_V5.*
    Directory SYS$COMMON:[SYSMGR]
    SYSTARTUP_V5.COM;1  SYSTARTUP_V5.TEMPLATE;1
    Total of 2 files.
    SYSMAN>
```

The SPAWN command in this example allows you to view a directory listing of some files in the SYS\$MANAGER directory. After the DIRECTORY command executes, control returns to the parent process.

```
2  SYSMAN> SPAWN
    $ EDIT SYSTARTUP_V5.COM
    .
    .
    $ LOGOUT
    Process SYSTEM_1 logged out at 28-JUN-1990 10:05:17.24
    SYSMAN>
```

This example shows how you can use the SPAWN command to leave SYSMAN and edit a file. The LOGOUT command returns you to SYSMAN.



```
3 SYSMAN> SPAWN /NOLOGICAL_NAMES SET HOST
  _Node: DUNBAR
      .
      .
      .
$ LOGOUT
%REM-S-END, control returned to node _NEWT::
SPAWN>
```

The example shows how you can use the SPAWN command to create a subprocess in which you can SET HOST. When you want to leave node DUNBAR, enter the LOGOUT command. The /NOLOGICAL\_NAMES qualifier prevents the logical names of the parent process from being copied to the subprocess.

---

## **STARTUP ADD**

Adds a component to the startup database.

Requires read (R) and write (W) access to the startup database.

---

**FORMAT**            **STARTUP ADD**    *FILE filespec*

---

**PARAMETER**    **FILE**  
Directs SYSMAN to add a component to the startup database. SYSMAN modifies STARTUP\$STARTUP\_LAYERED by default.

**filespec**  
Specifies which file to add to the startup database. Each component of the startup database must have a file type of COM or EXE and reside in SYS\$STARTUP.

---

**QUALIFIERS**    **/CONFIRM**  
**/NOCONFIRM (default)**  
Controls whether SYSMAN displays the file specification of each file before adding it to the startup database and requests you to confirm the addition. If you specify /CONFIRM, you must respond to the prompt with a Y (Yes) or a T (True) and press Return before the file is added. If you enter anything else, such as N or NO, the requested file is not added.

**/LOG**  
**/NOLOG**  
Controls whether the STARTUP ADD command displays the file specification of each file after it has been added.

**/MODE=mode**  
Specifies the mode of execution for the file. Valid modes include DIRECT, SPAWN, BATCH, or ANY , described in Section 4.6.

**/NODE=(node1,node2,...,nodex)**  
Names the nodes within the cluster that run the file during startup. By default, a startup file executes on all nodes in the cluster.

**/PARAMETER=(P1:arg1,P2:arg2,...,P8:arg8)**  
Specifies the parameters that are to be passed to the file during startup. Parameters that are omitted receive the default parameters defined by the system parameter STARTUP\_Pn.

**/PHASE=phase-name**  
Indicates the phase within system startup when the file is to be executed. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END. LPMAIN is the default.

---

**DESCRIPTION**

The **STARTUP ADD** command adds a component to the startup database. Startup components are the command procedures or executable files that perform actual startup work. Files from the startup database are used to start the VMS operating system, site-specific programs, and layered products. **STARTUP\$STARTUP\_VMS** and **STARTUP\$STARTUP\_LAYERED** list the components of the startup database.

Because a cluster typically shares one copy of the startup database, the **SYSMAN** environment can be defined as clustered or as a single node within the cluster.

---

**EXAMPLE**

```
SYSMAN> STARTUP ADD FILE /MODE=DIRECT /PHASE=LPMAIN -  
_SYSMAN> FOR$LPMAIN_043_STARTUP.COM
```

The **STARTUP ADD** command in this example adds a record to the startup database that starts **FORTRAN** Version 4.3.

**SYSMAN**  
**STARTUP DISABLE**

---

## **STARTUP DISABLE**

Prevents a file in the startup database from executing.  
Requires read (R) and write (W) access to the startup database.

---

**FORMAT**            **STARTUP DISABLE**    *FILE filespec*

---

**PARAMETER**    **FILE**  
Directs SYSMAN to disable a component of the startup database. SYSMAN modifies STARTUP\$STARTUP\_LAYERED by default.

**filespec**  
Specifies the name of a component in the startup database. The startup file must reside in SYS\$STARTUP and have a file type of COM or EXE. The wildcard characters % and \* are permitted.

---

**QUALIFIERS**    **/CONFIRM**  
**/NOCONFIRM (default)**  
Controls whether the STARTUP DISABLE command displays the file specification of each file before disabling it in the startup database and requests you to confirm that the file should be disabled. If you specify /CONFIRM, you must respond to the prompt with a Y (Yes) or a T (True) and press Return before the file is disabled. If you enter anything else, such as N or NO, the requested file is not disabled.

**/LOG**  
**/NOLOG**  
Controls whether the STARTUP DISABLE command displays the file specification of each file after it has been disabled.

**/NODE=(node1,node2,...,nodex)**  
Identifies nodes within the cluster that do not run the file during startup. By default, the startup file is disabled on all nodes in the cluster.

**/PHASE=phase-name**  
Indicates the phase of system startup in which the specified file normally executes. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END. LPMAIN is the default.

---

**DESCRIPTION**    The STARTUP DISABLE command prevents a file in the startup database from executing. The command edits a record in the startup database, temporarily disabling the file.

---

**EXAMPLE**

SYSMAN> STARTUP DISABLE FILE /NODE=BELA FOR\$LPMAIN\_043\_STARTUP.COM

The command in this example modifies the startup database so that FORTRAN will not be installed on node BELA.

# SYSMAN

## STARTUP ENABLE

---

## STARTUP ENABLE

Allows a previously disabled file in the startup database to execute during system startup.

Requires read (R) and write (W) access to the startup database.

---

**FORMAT**            **STARTUP ENABLE**    *FILE filespec*

---

**PARAMETER**    **FILE**  
Directs SYSMAN to enable a component of the startup database. SYSMAN modifies STARTUP\$STARTUP\_LAYERED by default.

**filespec**  
Specifies the name of the startup file that you are enabling. Wildcard characters are accepted.

---

**QUALIFIERS**    **/CONFIRM**  
**/NOCONFIRM (default)**  
Controls whether the STARTUP ENABLE command displays the file specification of each file before enabling it in the startup database and requests you to confirm that the file should be enabled. If you specify /CONFIRM, you must respond to the prompt with a Y (Yes) or a T (True) and press Return before the file is enabled. If you enter anything else, such as N or NO, the requested file is not enabled.

**/LOG**  
**/NOLOG**  
Controls whether the STARTUP ENABLE command displays the file specification of each file after it has been enabled.

**/NODE=(node1,node2,...,nodex)**  
Names nodes within the cluster where the file should be enabled. By default, the startup file is enabled on all nodes.

**/PHASE=phase-name**  
Indicates the phase within system startup when the specified file is to be enabled. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END. LPMAIN is the default.

---

**DESCRIPTION**    The STARTUP ENABLE command permits a file that was previously disabled to execute during system startup.

---

**EXAMPLE**

SYSMAN> STARTUP ENABLE FILE /NODE=ZURICH FOR\$LPMAIN\_043\_STARTUP.COM

The command in this example modifies the startup database. Node ZURICH will have FORTRAN Version 4.3 installed at startup.

# SYSMAN

## STARTUP MODIFY

---

# STARTUP MODIFY

Changes information associated with a startup file in the startup database.  
Requires read (R) and write (W) access to the startup database.

---

**FORMAT**            **STARTUP MODIFY**    *FILE filespec*

---

**PARAMETER**    **FILE**  
Directs SYSMAN to modify a record in the startup database. SYSMAN modifies STARTUP\$STARTUP\_LAYERED by default.

**filespec**  
Selects a startup file for modification. Wildcard characters are accepted.

---

**QUALIFIERS**    **/CONFIRM**  
**/NOCONFIRM (default)**  
Controls whether the STARTUP MODIFY command displays the file specification of each file before modifying its startup characteristics in the startup data file and requests you to confirm that the file characteristics should be modified. If you specify /CONFIRM, you must respond to the prompt with a Y (Yes) or a T (True) and press Return before the file is modified. If you enter anything else, such as N or NO, the requested file is not modified.

**/LOG**  
**/NOLOG**  
Controls whether the STARTUP MODIFY command displays the file specification of each file after its startup characteristics have been modified.

**/MODE=mode**  
Changes the mode of execution for a startup file. Valid modes include DIRECT, SPAWN, BATCH, or ANY , described in Section 4.6.

**/NAME=filespec**  
Changes the name of the startup file. The file must reside in SYS\$STARTUP.

**/PARAMETER=(P1:arg1,P2:arg2,...,P8:arg8)**  
Changes the parameters that are to be passed to the file during startup. Parameters that are omitted receive the default parameters defined by the system parameter STARTUP\_Pn.

**/PHASE=phase-name**  
Selects startup files for modification based on the phase in which they run. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END. LPMAIN is the default.



---

**DESCRIPTION**

The **STARTUP MODIFY** command edits startup information associated with components in the startup database. For example, the command can rename a file or change the parameters that are passed to a file during startup. You can select a group of files for modification based on the phase in which they run.

---

**EXAMPLE**

```
SYSMAN> STARTUP MODIFY FILE FOR$LPMAIN_043_STARTUP.COM -  
_SYSMAN> /PARAM=(P3:TRUE,P4:FALSE) /CONFIRM
```

The command in this example changes two startup parameters for the command procedure **FOR\$LPMAIN\_043\_STARTUP.COM**.

# SYSMAN

## STARTUP REMOVE

---

## STARTUP REMOVE

Removes a record in the startup database, so the specified startup file no longer executes during system startup.

Requires read (R) and write (W) access to the startup database.

---

**FORMAT**            **STARTUP REMOVE**    *FILE filespec*

---

**PARAMETER**    **FILE**  
Directs SYSMAN to remove a component from the startup database. SYSMAN modifies STARTUP\$STARTUP\_LAYERED by default.

**filespec**  
Specifies the name of the file to remove from the startup database. Wildcard characters are accepted.

---

**QUALIFIERS**    **/CONFIRM**  
**/NOCONFIRM (default)**  
Controls whether the STARTUP REMOVE command displays the file specification of each file before deleting its record in the startup database and requests you to confirm that the file should be deleted. If you specify /CONFIRM, you must respond to the prompt with a Y (Yes) or a T (True) and press Return before the file is removed. If you enter anything else, such as N or NO, the requested file is not removed.

**/LOG**  
**/NOLOG**  
Controls whether SYSMAN displays the file specification of each file after it has been removed.

**/PHASE=phase-name**  
Indicates the phase of system startup from which the file should be removed. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END.

---

## EXAMPLE

```
SYSMAN> STARTUP REMOVE FILE FOR$LPMAIN_043_STARTUP.COM /LOG
```

The command in this example takes the file FOR\$LPMAIN\_043\_STARTUP.COM out of the startup database.

---

## **STARTUP SET DATABASE**

Establishes the current startup database.

---

**FORMAT**            **STARTUP SET DATABASE**    *database*

---

**PARAMETER**        ***database***  
Specifies the name of the target database, which is `STARTUP$STARTUP_LAYERED` by default. The second database, `STARTUP$STARTUP_VMS` is available for viewing; however, Digital recommends that you do not modify it.

---

**QUALIFIERS**        *None.*

---

### **EXAMPLE**

```
SYSMAN> STARTUP SET DATABASE STARTUP$STARTUP_LAYERED
%SYSMAN-I-NEWCOMPFIL, current component file is now STARTUP$STARTUP_LAYERED
SYSMAN> STARTUP SHOW FILE
%SYSMAN-I-COMPFIL, contents of component database on node LUCERN
Phase    Mode    File
-----  -
LPBEGIN  DIRECT  VMS$LPBEGIN_050_STARTUP.COM
LPMAIN   DIRECT  FOR$LPMAIN_043_STARTUP.COM
```

The commands in this example establish the layered products database as the default, so it can be displayed.

---

## **STARTUP SHOW**

Displays the name of the current startup database or its components.

---

|               |                     |                                |
|---------------|---------------------|--------------------------------|
| <b>FORMAT</b> | <b>STARTUP SHOW</b> | <i>DATABASE</i><br><i>FILE</i> |
|---------------|---------------------|--------------------------------|

---

|                  |                 |
|------------------|-----------------|
| <b>PARAMETER</b> | <b>DATABASE</b> |
|------------------|-----------------|

Directs SYSMAN to display the name of the current startup database. There are two startup databases: STARTUP\$STARTUP\_LAYERED and STARTUP\$STARTUP\_VMS. Digital recommends that you do not modify the STARTUP\$STARTUP\_VMS database.

**FILE**

Displays the contents of the current startup database. The display includes the file name, phase, and mode of execution for each component in the database.

---

|                   |              |
|-------------------|--------------|
| <b>QUALIFIERS</b> | <b>/FULL</b> |
|-------------------|--------------|

Displays full information about each component in the database. In addition to the phase, file name, and mode of execution for each startup component, SYSMAN displays the nodes on which the file executes and the parameters passed to the file. Relevant with the FILE parameter.

**/NODE**

Displays the nodes within the cluster on which the file executes. By default, a startup file executes on all nodes in an environment. Relevant with the FILE parameter.

**/OUTPUT=filespec**

Redirects command output from SYS\$OUTPUT to the file named with the qualifier. Without a **filespec**, SYSMAN writes the output to SYSMAN.LIS in the current directory.

**/PARAMETERS**

Lists the parameters with which the startup file executes. Parameters that are not specified receive the defaults defined by the system parameter STARTUP\_Pn. Relevant with the FILE parameter.

**/PHASE=phase-name**

Displays components that execute in a specific phase of system startup. Valid phases include LPBEGIN, LPMAIN, LPBETA, and END. LPMAIN is the default. Relevant with the FILE parameter.

---

**EXAMPLE**

```
SYSMAN> STARTUP SET DATABASE STARTUP$STARTUP_VMS
SYSMAN> STARTUP SHOW FILE
%SYSMAN-I-COMPFIL, contents of component database on node LUCERN
Phase      Mode      File
-----
BASEENVIRON DIRECT  VMS$BASEENVIRON_050_LIB.COM
BASEENVIRON CALLED  VMS$BASEENVIRON_050_SMISERVER.COM
BASEENVIRON DIRECT  VMS$BASEENVIRON_050_VMS.COM
.
.
.
```

The commands in this example display the contents of the VMS startup database.

---

## **SYS\_LOADABLE ADD**

Adds an entry in the system images file SYS\$UPDATE:VMS\$SYSTEM\_IMAGES.IDX.

**Caution:** The **SYS\_LOADABLE ADD** command is not intended for general use. Only advanced system programmers should use this command.

---

**FORMAT**            **SYS\_LOADABLE ADD** *product image*

---

**PARAMETER**    ***product***  
A 1- to 8-character product mnemonic that uniquely identifies a loadable image. For user-written images, this should typically contain the string *\_LOCAL\_*.

***image***  
The file name of the system loadable image you want to add. A file name is the only value you can specify for this parameter. Do not specify a device, directory, file type, or wildcard characters.

---

**QUALIFIERS**    ***/LOAD\_STEP***  
Indicates the step of the booting process at which you want the image loaded. Valid load steps are INIT (which causes the the system initialization code to load the image), and SYSINIT (which causes the SYSINIT process to load the image).

If you do not specify a value for the */LOAD\_STEP* qualifier, it defaults to SYSINIT.

***/[NO]LOG***  
Controls whether the SYS\_LOADABLE ADD command displays a notification after the entry has been added.

***/MESSAGE***  
Lets you specify the text of a message that is displayed when the appropriate condition is met (see the */SEVERITY* qualifier). The default message is "system image load failed".

***/SEVERITY***  
Determines how the image load status will affect console output and booting progress. You can specify the following values for this qualifier:

- **FATAL**—If an error occurs loading the image, display the error message and BUGCHECK information.
- **INFORMATION**—Display the message and continue processing.

## SYSMAN SYS\_LOADABLE ADD

- **SUCCESS**—Continue even if there is an error loading the image. Does not display the message.
- **WARNING**—If an error occurs loading the image, display the error message and continue processing.

If you do not specify a value for the /SEVERITY qualifier, it defaults to WARNING.

---

### DESCRIPTION

The SYS\_LOADABLE ADD command adds an entry to the system images file SYS\$UPDATE:VMS\$SYSTEM\_IMAGES.IDX. You can then process this file using the command procedure SYS\$UPDATE:VMS\$SYSTEM\_IMAGES.COM. Processing the file with VMS\$SYSTEM\_IMAGES.COM generates a new system images data file that the system uses when it boots. See the *VMS Version 5.4 New Features Manual* for a complete description of adding user-written system images.

If the file SYS\$UPDATE:VMS\$SYSTEM\_IMAGES.IDX does not exist, the SYS\_LOADABLE ADD command creates a new one.





---

# Index

---

---

## A

---

ALF (auto-login facility) commands • SM-13

See also Auto login facility

Alias cluster name • SM-4

ATTACH command • SM-27

Attaching to a process • SM-13

Auto-login facility • SM-13

adding records • SM-23

---

---

## B

---

Batch execution

as startup mode • SM-12

on remote nodes • SM-19

---

---

## C

---

Cluster

command execution • SM-52

defining environment • SM-4

displaying group number • SM-33

displaying multicast address • SM-33

displaying time • SM-34

executing commands • SM-4

group number • SM-29

local and nonlocal • SM-4

management • SM-2

password • SM-29

setting time • SM-31

updating security data • SM-29

CLUSTER\_AUTHORIZE.DAT • SM-29, SM-33

CLUSTER\_CONFIG.COM • SM-29

Command execution • SM-4

on remote nodes • SM-14

Command procedure

correcting system time • SM-31

executing • SM-21

executing at system startup • SM-80

executing DCL commands with DO • SM-8,  
SM-52

executing in SYSMAN • SM-8

---

Command procedure (Cont.)

for system startup • SM-1, SM-11

LOGIN.COM • SM-5

Command summary • SM-15

CONFIGURATION commands • SM-10

See also Local area cluster

---

---

## D

---

DCL command • SM-8

executing with SYSMAN DO • SM-8

DECnet

using to manage remote nodes • SM-2

Defining keys • SM-14, SM-35

Device

displaying the default • SM-74

logical name translation • SM-40

resetting the default • SM-6

setting defaults for file specifications • SM-70

specifying quotas for • SM-38

DIRECT mode • SM-12

Directory default

displaying • SM-74

modifying • SM-70

resetting • SM-6

DIRECTORY/SIZE command • SM-9

Disk quota

adding entry • SM-38

creating • SM-40

disabling • SM-10, SM-43

displaying • SM-9, SM-50

establishing • SM-9

file • SM-8

initializing usage count • SM-38

maintaining • SM-10

modifying an entry • SM-45

rebuilding • SM-47

removing an entry • SM-42, SM-48

Disk Quota (DISKQUOTA) Utility • SM-8

See also DISKQUOTA commands

DISKQUOTA commands • SM-1, SM-8

See also Disk Quota • SM-38

DISKQUOTA DELETE command

See DISKQUOTA REMOVE

Disk usage • SM-9

DO command • SM-8, SM-52

---

## Index

Dynamic parameter  
  modifying • SM-67

---

## E

---

END phase • SM-88  
Environment  
  See Management environment  
Executing a command procedure • SM-21  
Executing SYSMAN command procedures • SM-8  
Executive loaded images  
  See also System loadable image  
EXIT command • SM-54  
Exiting from SYSMAN • SM-18

---

## F

---

File  
  quota • SM-8  
    See also Quota file  
  QUOTA.SYS • SM-9  
  startup • SM-11, SM-80, SM-82, SM-90  
  STARTUP.COM • SM-11  
  SYSMAN.LIS • SM-26, SM-33, SM-34, SM-50,  
    SM-52, SM-53, SM-63, SM-90

---

## G

---

Group number • SM-29  
  displaying • SM-33

---

## H

---

HELP command • SM-55

---

## I

---

Image  
  See also System loadable image  
Initialization file  
  file specification • SM-14  
  use with SYSMAN • SM-14

---

Index-2

Input stream  
  switching control to other processes • SM-27  
Invoking SYSMAN • SM-18  
  privilege requirements • SM-18

---

## K

---

Keys  
  defining • SM-14, SM-35  
  displaying definitions • SM-73

---

## L

---

License  
  activating with SYSMAN • SM-56  
  deactivating with SYSMAN • SM-57  
  manipulating with SYSMAN • SM-12  
LICENSE commands • SM-12  
  See also License  
Local area cluster  
  displaying network information • SM-33  
  displaying system date and time • SM-34  
  modifying security data • SM-29  
  security • SM-10  
  setting system date and time • SM-31  
Local node • SM-2  
LPBEGIN phase • SM-88  
LPBETA phase • SM-88  
LPMAIN phase • SM-88

---

## M

---

Management environment  
  centralized • SM-4  
  clusterwide • SM-4  
  defining • SM-2 to SM-4, SM-68  
  individual nodes • SM-3  
  local and nonlocal environments • SM-3  
MCR command • SM-18, SM-52  
Mode of execution • SM-12, SM-80  
MOUNT/CLUSTER command • SM-8

---

---

## O

---

Overdraft limit • SM-9

---

## P

---

### Parameter

See System parameter

### Parameter file

writing • SM-67

PARAMETERS commands • SM-1, SM-11

See also System parameters

Password • SM-5, SM-68

conditions requiring • SM-3

for local area cluster • SM-29

Permanent quota • SM-9

### Privilege

changing current • SM-6

displaying current • SM-74

modifying for environment • SM-70

required by SYSMAN • SM-18

### Process

#### hibernation

with ATTACH command • SM-27

switching control of input stream to • SM-27

Profile • SM-54, SM-68

adjusting • SM-5, SM-70

changing default directory • SM-6

current privileges • SM-5

displaying • SM-74

---

## Q

---

QUOTA.SYS • SM-9

Quota file • SM-8

adding an entry • SM-38

creating • SM-40

disabling • SM-10, SM-43

displaying • SM-50

enabling • SM-44

modifying • SM-45

removing an entry • SM-42, SM-48

updating • SM-10, SM-47

valid entries • SM-38

---

## R

---

### Range check

disabling • SM-58

enabling • SM-59

Remote node • SM-2

REPLY command • SM-19

Restriction of SYSMAN • SM-18

Rights identifier • SM-6, SM-38, SM-45

---

## S

---

### Security

See also Local area cluster

in local area cluster • SM-10

managing remote nodes • SM-6

SET ENVIRONMENT command • SM-3, SM-68

SET PROFILE command • SM-6, SM-70

SET QUORUM/CLUSTER command • SM-8

SET TIMEOUT command • SM-4, SM-71

SHOW ENVIRONMENT command • SM-3, SM-68, SM-72

SHOW KEY command • SM-73

SHOW PROFILE command • SM-6, SM-74

SHOW TIMEOUT command • SM-75

SMISERVER process

attributes of • SM-5

starting • SM-14

SPAWN command • SM-76

and ATTACH command • SM-27

Spawning a subprocess • SM-13

STARTUP\$STARTUP\_LAYERED • SM-11, SM-89

STARTUP\$STARTUP\_VMS • SM-11, SM-89

STARTUP commands • SM-1, SM-11

See also Startup database

Startup database

adding a component • SM-81

deleting a record • SM-88

disabling a component • SM-82

displaying contents • SM-90

enabling a component • SM-84

modifying records • SM-86

setting the default • SM-89

Startup phase • SM-11, SM-80

STARTUP\_Pn parameter • SM-80

Subprocess

See also SPAWN command

## Index

### Subprocess (Cont.)

- creating with SPAWN command • SM-76
- switching control of input stream to • SM-27

SYS\$STARTUP • SM-11

SYSGEN parameter • SM-11

SYSMAN command • SM-18

SYSMANINI logical name • SM-14

SYSMAN Utility • SM-1

- command procedures • SM-8
- commands • SM-15
- directing output • SM-52
- executing SYSMAN commands • SM-14
- exiting from • SM-18
- invoking • SM-18
- managing
  - See System management
- privilege requirements • SM-18
- restrictions • SM-18

System disk • SM-8

System loadable image

- adding • SM-12, SM-92
- removing • SM-12, SM-94

System management

- authorization checks • SM-6
- creating disk quota files • SM-40
- defining a cluster • SM-4
- defining target system • SM-68
- displaying target environment • SM-72
- displaying time • SM-34
- displaying user profile • SM-74
- displaying wait time for node response • SM-75
- executing DCL commands • SM-52
- how commands execute • SM-4
- limiting response time • SM-71
- local and nonlocal • SM-3
- maintaining disk quotas
  - See Disk quota
- modifying user profile • SM-70
- parameters

- See System parameter
- setting system time • SM-31
- setting time • SM-10
- setting wait time for node response • SM-71
- startup procedures

See Startup database

System Management (SYSMAN) Utility

See SYSMAN Utility

System management environment

See Management environment

System parameters

- checking • SM-59
- disabling checking • SM-58
- displaying • SM-10, SM-62
- manipulating • SM-10, SM-66
- modifying dynamic • SM-10, SM-67
- setting • SM-10, SM-60
- work area • SM-66, SM-67
- writing contents of • SM-67

System time

- displaying • SM-34
- setting • SM-10, SM-31

SYS\_LOADABLE commands • SM-12

---

## T

Time

- displaying • SM-34
- setting system • SM-10, SM-31
- updating in a cluster • SM-32

Timeout period • SM-4

- displaying • SM-75
- setting • SM-71

---

## U

UAF (user authorization file) • SM-5

UIC (user identification code) • SM-38

UIC [0,0] • SM-9

Usage count

- DIRECTORY/SIZE command • SM-9
- DISKQUOTA display • SM-9
- updating • SM-47

USE command • SM-8

---

## V

Validation of users • SM-5

Volume quota file

See Disk quota

---

# W

---

Work area

copying to parameter file • SM-67

loading parameters • SM-66



## How to Order Additional Documentation

---

### Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

### Electronic Orders

To place an order at the Electronic Store, dial 800-DEC-DEMO (800-332-3366) using a 1200- or 2400-baud modem. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

### Telephone and Direct Mail Orders

| <b>Your Location</b>                  | <b>Call</b>  | <b>Contact</b>  |
|---------------------------------------|--------------|---|
| Continental USA,<br>Alaska, or Hawaii | 800-DIGITAL  | Digital Equipment Corporation<br>P.O. Box CS2008<br>Nashua, New Hampshire 03061   |
| Puerto Rico                           | 809-754-7575 | Local Digital subsidiary  |
| Canada                                | 800-267-6215 | Digital Equipment of Canada<br>Attn: DECdirect Operations KAO2/2<br>P.O. Box 13000<br>100 Herzberg Road<br>Kanata, Ontario, Canada K2K 2A6                |
| International                         | _____        | Local Digital subsidiary or<br>approved distributor   |
| Internal <sup>1</sup>                 | _____        | USASSB Order Processing - WMO/E15<br><i>or</i><br>U.S. Area Software Supply Business<br>Digital Equipment Corporation<br>Westminster, Massachusetts 01473 |

---

<sup>1</sup>For internal orders, you must submit an Internal Software Order Form (EN-01740-07).







# Reader's Comments

VMS SYSMAN Utility  
Manual  
AA-LA26B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

| I rate this manual's:                      | Excellent                | Good                     | Fair                     | Poor                     |
|--|--------------------------|--------------------------|--------------------------|--------------------------|
| Accuracy (software works as manual says)   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Completeness (enough information)          | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Clarity (easy to understand)               | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Organization (structure of subject matter) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Figures (useful)                           | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Examples (useful)                          | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Index (ability to find topic)              | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Page layout (easy to find information)     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

I would like to see more/less \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

What I like best about this manual is \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

What I like least about this manual is \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

I found the following errors in this manual:

| Page  | Description |
|-------|-------------|
| _____ | _____       |
| _____ | _____       |
| _____ | _____       |
| _____ | _____       |
| _____ | _____       |

Additional comments or suggestions to improve this manual:  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

I am using **Version** \_\_\_\_\_ of the software this manual describes.

Name/Title \_\_\_\_\_ Dept. \_\_\_\_\_  
Company \_\_\_\_\_ Date \_\_\_\_\_  
Mailing Address \_\_\_\_\_  
\_\_\_\_\_ Phone \_\_\_\_\_



--- Do Not Tear - Fold Here and Tape ---

**digital**<sup>TM</sup>

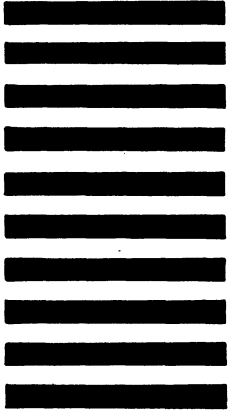


No Postage  
Necessary  
if Mailed  
in the  
United States

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION  
Corporate User Publications—Spit Brook  
ZK01-3/J35 110 SPIT BROOK ROAD  
NASHUA, NH 03062-9987



--- Do Not Tear - Fold Here ---

Cut Along Dotted Line

# Reader's Comments

VMS SYSMAN Utility  
Manual  
AA-LA26B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

| I rate this manual's:                      | Excellent                | Good                     | Fair                     | Poor                     |
|--|--------------------------|--------------------------|--------------------------|--------------------------|
| Accuracy (software works as manual says)   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Completeness (enough information)          | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Clarity (easy to understand)               | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Organization (structure of subject matter) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Figures (useful)                           | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Examples (useful)                          | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Index (ability to find topic)              | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Page layout (easy to find information)     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

I would like to see more/less \_\_\_\_\_

What I like best about this manual is \_\_\_\_\_

What I like least about this manual is \_\_\_\_\_

I found the following errors in this manual:

| Page  | Description |
|-------|-------------|
| _____ | _____       |
| _____ | _____       |
| _____ | _____       |
| _____ | _____       |

Additional comments or suggestions to improve this manual:

I am using Version \_\_\_\_\_ of the software this manual describes.

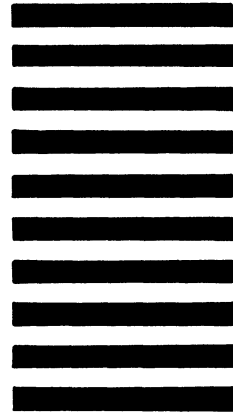
Name/Title \_\_\_\_\_ Dept. \_\_\_\_\_  
Company \_\_\_\_\_ Date \_\_\_\_\_  
Mailing Address \_\_\_\_\_  
Phone \_\_\_\_\_

--- Do Not Tear - Fold Here and Tape ---

**digital**<sup>TM</sup>



No Postage  
Necessary  
if Mailed  
in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION  
Corporate User Publications—Spit Brook  
ZK01-3/J35 110 SPIT BROOK ROAD  
NASHUA, NH 03062-9987



--- Do Not Tear - Fold Here ---

Cut Along Dotted Line