

VSI Job Management for OpenVMS Administration Guide

Operating System and Version: VSI OpenVMS x86-64 Version 9.2-3

Software Version: VSI Job Management V3.1 for OpenVMS

VSI Job Management for OpenVMS Administration Guide



VMS Software

Copyright © 2025 VMS Software, Inc. (VSI), Boston, Massachusetts, USA

Legal Notice

Confidential computer software. Valid license from VSI required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for VSI products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. VSI shall not be liable for technical or editorial errors or omissions contained herein.

All other trademarks and registered trademarks mentioned in this document are the property of their respective holders.

Table of Contents

Preface	ix
1. About VSI	ix
2. Document Structure	ix
3. Related Documents	ix
4. VSI Encourages Your Comments	x
5. OpenVMS Documentation	x
Chapter 1. An Overview of Job Management Manager	1
1.1. About Job Management Manager	1
1.2. Features and Uses	1
1.3. How to Access Job Management Manager	2
1.3.1. DCL interface (Chapter 4)	2
1.3.2. Motif interface (Chapter 5)	2
1.4. Comparing Job Management Manager Detached Processes with OpenVMS Standard Batch Processes	2
1.5. Privileges Needed to Use Job Management Manager	3
1.5.1. Privileges for Management Commands	4
1.6. Wide Area Network Support	5
Chapter 2. How Job Management Manager Works	7
2.1. Overview	7
2.1.1. Starting and Stopping Job Management Manager	7
2.2. Automating and Managing Jobs	7
2.2.1. Job, Job Database, and Job Record	7
2.3. How Job Management Manager Processes Jobs	9
2.4. How Job Management Manager Reports a Completed Job	9
2.5. Job States	10
2.5.1. Restarting Jobs	11
2.6. Job Database Listing	11
2.6.1. Specifying a Job	12
2.7. Job Output Files	13
2.8. Using Wildcards	13
2.9. Accessing Jobs on Remote Nodes	13
2.10. Prefunctions and Postfunctions	14
2.10.1. Prefunction and Postfunction Processing Rules	14
2.11. Job Dependencies	15
2.11.1. Remote Nodes	15
2.12. Delta Time Formats	16
2.12.1. OpenVMS Delta Time	16
2.12.1.1. Rules for Formatting OpenVMS Delta Time	17
2.12.2. Job Management Manager Delta Time	17
2.12.2.1. Rules for Formatting Job Management Manager Delta Time	17
2.12.2.2. Time Creep in Jobs	18
2.13. Special Days	18
2.13.1. How Special Days Affect Run Times	19
2.13.1.1. Dynamic Changes to Special Day Classes	19
2.13.2. Creating the First Special Day Class	20
2.13.3. /SD_ACTION Qualifier	20
2.14. Using Batch Mode	20
2.14.1. Job Management Manager Batch Characteristics	20
2.14.2. Job Management Manager Batch Execution	21

2.14.3. Batch Mode Logicals	22
2.15. Job Management Manager in an OpenVMS Cluster	22
2.15.1. How the Default Node Runs a Job Management Manager Job	23
2.15.2. Supporting Batch Mode Jobs in an OpenVMS Cluster	23
2.15.3. Balancing Job Loads Among Nodes in an OpenVMS Cluster	23
2.15.4. Recovering Jobs from Nodes That Fail	24
Chapter 3. Getting Started with Job Management Manager	25
3.1. Job Management Manager Overview	25
3.2. Roles and Responsibilities	25
3.3. User Interfaces	25
3.4. Steps Leading Toward Automation	26
3.5. Controlling Job Access	26
3.5.1. Identifier-Based Access to Jobs	27
3.5.2. Mixed Environments	27
3.6. The Callable Interface (SDK)	29
3.7. System Management and Normal Maintenance	30
3.7.1. Dependent Jobs Relying on a Single Condition	31
3.7.2. Synchronizing Local and Remote Databases	32
3.7.3. Activating the Debug Log File	32
3.8. Load Balancing	33
3.8.1. Rating a System	33
3.8.2. How Load Balance Groups Work	34
3.9. Job Management Manager/Agent Performance Tuning	37
Chapter 4. Using the DCL Command Interface	39
4.1. Overview	39
4.2. Accessing Job Management Manager and Entering Commands	39
4.2.1. Is Job Management Manager Installed?	40
4.2.2. Using Help	40
4.3. General and Management Commands	40
4.3.1. General Command	40
4.3.2. Management Command	41
4.4. Creating a Job Management Manager Job	41
4.5. Creating Job Dependencies	42
4.5.1. Job-Dependency Restrictions	43
4.5.2. Job-Dependency Processing Rule	43
4.5.2.1. Overriding Job Dependencies	43
4.6. Assigning a Job to a Group or Type	43
4.7. Special Days	44
4.7.1. Defining Special Day Classes	44
4.7.2. Restriction Expressions: Relating Special Days to Jobs	45
4.7.3. Special Day Actions	46
4.7.4. Showing Information Related to Special Days	47
4.8. Sending OPCOM Messages	47
4.9. Creating a Job in Batch Mode	47
4.10. Summary of Job Management Manager Commands	48
4.10.1. General Commands	48
4.10.2. Management Commands	49
Chapter 5. Using the Motif Interface	51
5.1. Overview	51
5.1.1. Changing the Resource Files	52
5.1.2. Motif Interface	53

5.2. Getting Started with the Motif Interface	53
5.2.1. Using Mouse Buttons	53
5.2.2. Using Menus and Buttons	53
5.2.3. Starting Job Management Manager with the Motif Interface	54
5.2.4. The Load Jobs Dialog Box	55
5.2.5. Job Management Manager Graphical Display Window	56
5.2.6. Job Management Manager Linear Display Window	58
5.2.7. Job Management Manager Real Time Message Display Window	59
5.3. Interpreting Your Job Display	60
5.3.1. Symbols in Job Boxes	60
5.3.2. Selected and Unselected Job Boxes	61
5.3.3. Job Dependency Arrows	63
5.4. Loading Jobs to Display	63
5.5. Clearing and Changing the Currently Displayed Jobs	64
5.6. Adding to the Currently Displayed Job	65
5.7. Locating Jobs by Scrolling	65
5.8. Viewing Job Information	66
5.8.1. Viewing a Job's Attributes	66
5.8.2. Viewing Job Output Files	67
5.8.3. Viewing Job Resource Usage	67
5.8.3.1. While the Job Is Running	68
5.8.3.2. When a Job Exceeds Historical Usage	68
5.9. Creating a Job	69
5.10. Creating a Job Dependency	71
5.11. Deleting a Job Dependency	72
5.12. Controlling a Job: Aborting, Holding, Running, and Releasing	72
5.13. Editing a Job	72
5.14. Deleting a Job	74
5.15. Sorting Jobs in the Linear Display Window	74
5.16. Creating a Graphics File of Your Display	76
5.16.1. Using Navigation Windows	76
5.16.1.1. Graphical Display Window and Its Navigation Window	77
5.16.1.2. Linear Display Window and Its Navigation Window	78
5.16.2. Selecting and Displaying a New Portion of a Display Window	78
5.16.3. Resizing a Portion of the Graphical Display Window	78
5.17. Special Day Classes	79
5.17.1. Creating a Special Day Class	79
5.17.2. Editing a Special Day Class	80
5.17.3. Deleting a Special Day Class	81
5.17.4. Setting or Editing Day Class Restrictions	81
5.18. Load Balance Groups	82
5.18.1. The Load Balance Group Editor	83
5.18.2. Creating a Load Balance Group	83
5.18.3. Editing a Load Balance Group	84
5.18.4. Deleting a Load Balance Group	84
5.18.5. Setting or Editing a Load Balance Group for a Job	84
5.19. Exiting from the Motif Interface	85
Appendix A. Job Management Manager Components	87
A.1. Components	87
A.1.1. Files	87
A.1.2. System Components	87

Appendix B. Logical Names, Global Symbols, and Event Logs	89
B.1. Overview	89
B.2. Job Management Manager Logical Names	89
B.3. Run Time Global Symbols Available to Jobs	92
B.4. Writing Job Events to a Log File	94
Appendix C. Customizing Load Balancing	97
C.1. Overview	97
C.2. Customizing Load Balancing	97
C.3. Restricting a Job to Run on a Specific Node	98
Appendix D. Using Extended Job Fields	99
D.1. General Best Practices	99
D.2. Prompting Best Practices	99
D.3. Script Best Practices	100
Appendix E. Troubleshooting and FAQs	101
E.1. Troubleshooting the Manager	101
E.1.1. Next Run Time Is NEVER	101
E.1.2. Next Run Time Is Wrong	101
E.1.3. Job Did Not Run or Did Not Run at Specified Time	101
E.1.4. Job Runs Unexpectedly	102
E.1.5. Notification of Special Days Restriction on Job and It Still Runs	103
E.1.6. Cannot Access Job by Name	103
E.1.7. Job Fails Immediately at Run Time	103
E.1.8. Job is Missing	103
E.1.9. Output File Is Not Being Written	104
E.1.10. No Notification Upon Completion	104
E.1.11. No Mail Upon Completion	104
E.1.12. No OPCOM Upon Completion	104
E.1.13. Job Aborts	104
E.1.14. Job Successful But Shows Termination Status as Failure	105
E.1.15. Job Management Manager Is Slow in Responding	105
E.1.16. Load Balancing Is Not Working	105
E.1.17. Jobs Going into Waiting State	106
E.1.18. Jobs Unexpectedly on HOLD	106
E.1.19. SCHED_REMOTE Will Not Start	106
E.1.20. SCHED\$LISTENER Will Not Start	106
E.1.21. Job Status = NOT RUNNING	106
E.1.22. Job in BATCH Mode Does Not Show in SCHEDULE SHOW JOB/BRIEF	107
E.1.23. DECwindows User Interface Results in Xtoolkit Error: Can't Open Display	107
E.1.24. NSCHED Process Goes into RWMBX State	107
E.1.25. Recreating Deleted Database	107
E.1.26. Repairing Job Records Containing Stale Cluster Aliases	108
E.1.26.1. Using NSCHED\$ALIAS_GROUPS to repair cluster aliases	108
E.1.26.2. Sample scenarios	109
E.2. Troubleshooting Job Management Agent	111
E.2.1. Job Management Manager Runs a Job on an Agent	111
E.2.2. Missing Information in the .ERR File	112
E.2.3. Proxy Validation Error When Connecting to a Remote Node	112
E.2.4. SYS\$NODE Incorrectly Defined	113
E.2.5. Agent Error Codes	113
E.3. Frequently Asked Questions	114

E.3.1. General Job Management Manager Questions	114
E.3.2. Setting Up Jobs	115
E.3.2.1. Examples	116
E.3.3. Mail and Broadcast Messages	117
E.3.4. Stopped Jobs	118
E.3.5. Reports	119
Glossary	121

Preface

1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

2. Document Structure

This manual is organized as follows:

- *Chapter 1, "An Overview of Job Management Manager"* explains its role in automating and managing jobs in OpenVMS
- *Chapter 2, "How Job Management Manager Works"* describes how the system automates, processes, and reports jobs.
- *Chapter 3, "Getting Started with Job Management Manager"* introduces the system's setup, roles, interfaces, automation steps, and job security management.
- *Chapter 4, "Using the DCL Command Interface"* describes how to use Job Management Manager with DCL commands.
- *Chapter 5, "Using the Motif Interface"* covers how to interact with Job Management Manager through the Motif interface.
- *Appendix A, "Job Management Manager Components"* outlines the key files and system components.
- *Appendix B, "Logical Names, Global Symbols, and Event Logs"* provides a list of Job Management Manager logical names and run time global symbols.
- *Appendix C, "Customizing Load Balancing"* provides information on customizing your own load balancing algorithm.
- *Appendix D, "Using Extended Job Fields"* demonstrates best practices to specify job fields that are longer than a standard OpenVMS command token.
- *Appendix E, "Troubleshooting and FAQs"* provides troubleshooting tips and answers to frequently asked questions for Job Management Manager and Job Management Agent.

3. Related Documents

This guide is part of the VSI Job Management for OpenVMS documentation set, which also includes the following:

- [VSI Job Management for OpenVMS Release Notes \[https://docs.vmssoftware.com/vsi-job-management-release-notes-v31/\]](https://docs.vmssoftware.com/vsi-job-management-release-notes-v31/)
- [VSI Job Management for OpenVMS Installation Guide \[https://docs.vmssoftware.com/vsi-job-management-installation-guide-v31/\]](https://docs.vmssoftware.com/vsi-job-management-installation-guide-v31/)
- [VSI Job Management for OpenVMS Command Reference Guide \[https://docs.vmssoftware.com/vsi-job-management-command-reference-guide-v31/\]](https://docs.vmssoftware.com/vsi-job-management-command-reference-guide-v31/)

- *VSI Job Management for OpenVMS Programming Guide* [<https://docs.vmssoftware.com/vsi-job-management-programming-guide-v31/>]

4. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: <docinfo@vmssoftware.com>. Users who have VSI OpenVMS support contracts through VSI can contact <support@vmssoftware.com> for help with this product.

5. OpenVMS Documentation

The full VSI OpenVMS documentation set can be found on the VMS Software Documentation webpage at <https://docs.vmssoftware.com>.

Chapter 1. An Overview of Job Management Manager

1.1. About Job Management Manager

The manager is a combination of job controller, database, and logging system. It helps data center personnel, individual users, and general OpenVMS users increase their productivity by providing a hands-off approach to running repetitive jobs.

Using Job Management Manager, you can perform the following tasks:

- Automate and manage jobs that run repeatedly, such as fiscal reports, operating system maintenance tasks, and user applications.
- Separate all scheduling specifics such as the schedule interval, completion notification, and job dependencies from the job's main task.
- Capture events using logging system, so you can generate accounting and historical reports.
- Is a fully distributed product. All functions and features work remotely as well as locally, transparent to the user except where noted.
- Provides for automatic node failover and easily customized load balancing for OpenVMS Cluster systems.

1.2. Features and Uses

The manager has a number of useful features for creating and managing repetitive jobs.

Jobs

- You can create jobs that consist of a single system-level command or a command file containing many commands.
- Your repetitive jobs run automatically, based on scheduled intervals and other criteria you specify. For example, you can specify mail notification for completed jobs, or what to do if a system failure occurs.
- You can modify selected job criteria without having to resubmit a job. The manager maintains your job criteria in its job database, separate from job actions.
- You can modify job characteristics for a single run or for all future runs.
- You can specify commands to run before or after a job (prefunctions and postfunctions).

Networks and Synchronized Jobs

- You can schedule jobs within a cluster of nodes or among remote nodes on a network.
- You can synchronize a job with up to 16 other jobs, in the same OpenVMS Cluster or on remote nodes. Those 16 jobs can be synchronized with other jobs, giving the effect of unlimited job dependencies.

Mail Notification and Reports

- You can send job completion notifications by OpenVMS mail, terminal broadcast messages, and OPCOM messages.
- The manager provides a standard format for log reports. You can customize the format to fit your specific requirements.
- You can view a job history for any number of jobs.

System Management

- You can restart interrupted jobs on failover nodes, based on the job criteria, restart labels, and strings within the command procedure.
- Management commands allow the system manager to control the Job Management Manager processing environment, including load balancing, failover, event logging, and network events.

1.3. How to Access Job Management Manager

There are two ways to access the manager:

1.3.1. DCL interface (Chapter 4)

DCL commands provide easy access on all OpenVMS hardware platforms to job database information and current job status. You can run, modify, or review jobs by job name, number, group, owner, or type. The manager supports the use of wildcards on each of a job's fields, so you can enter a single command for multiple jobs.

You can use the DCL interface to access the full set of Job Management Manager commands (general user commands and also management commands) if you have the required system privileges.

1.3.2. Motif interface (Chapter 5)

If you have a workstation or other DECwindows display device you can use the Motif interface, which provides you with a graphic view of Job Management Manager jobs. Jobs are represented by boxes. Job dependencies are indicated by arrows that show you the order in which jobs are performed.

You can access all general user commands using menus and dialog boxes; you cannot access management commands using the Motif interface.

1.4. Comparing Job Management Manager Detached Processes with OpenVMS Standard Batch Processes

The manager offers both detached and batch processing. For many repetitive tasks, Job Management Manager detached processing can be easier to use than OpenVMS standard batch processing. The table below shows how detached processing and OpenVMS batch (without Job Management Manager) handle several common tasks.

Function	Job Management Manager Detached Process	OpenVMS Batch Process (without Job Management Manager)
Processing a single DCL command	Allows single DCL commands or command files.	Requires a command file.
Modifying a job record	Modifies and runs jobs repeatedly.	Jobs must be resubmitted.
Modifying job criteria	Separates criteria from the job itself; you can modify criteria without modifying the job.	The job and its criteria must be deleted, modified, and resubmitted.
Identifying file versions	Uses a file name. Runs the most current file version or the version you specify.	Uses an OpenVMS record management services (RMS) file identification. Runs the version that existed when the job was submitted, regardless of any new changes made to the file.

Here is a list of some tasks you can perform using the manager that you cannot perform using standard OpenVMS batch mode:

- Override job settings for a single run
- Synchronize multiple jobs on an OpenVMS Cluster
- Specify failover actions for interrupted jobs
- Access jobs by Motif interfaces
- Specify prefunctions and postfunctions to perform before or after a job
- Issue warning messages when a job exceeds the maximum run time or does not start when expected

For functions that involve timing between different nodes, such as maximum run time and start time for remote or batch mode jobs, system times must be synchronized.

Note

OpenVMS detached processes created by the manager carry the processing mode of OTHER for detached processes, or BATCH for batch processes. Before you run existing command procedures under the manager, change the test for F\$MODE() .EQS. "BATCH", to:

```
IF F$MODE() .EQS. "BATCH" .OR. F$MODE() .EQS. "OTHER"
```

Batch mode further increases the versatility and usefulness of the manager. For a description of the manager, see Using Batch Mode in *Chapter 2, "How Job Management Manager Works"*.

1.5. Privileges Needed to Use Job Management Manager

The manager provides commands to create and control jobs, and management commands to control the operation of the manager system itself. In general, you need either SYSPRV or OPER OpenVMS privileges to issue the manager management commands (some management commands require other

privileges as well). General users can perform operations on their own jobs, and on other jobs based on OpenVMS privileges and identifiers.

There are three types of access that users can have to jobs:

- READ access permits you to see information about a job and to synchronize other jobs to that job.
- EXECUTE access allows you to control how a job runs, but not what it does.
- WRITE access allows full control of job characteristics.

Your OpenVMS privileges and your UIC, the OpenVMS identifiers and UIC associated with the job, and identifiers that you hold, determine the access that you have to a job.

You have READ access to a job if at least one of the following is true:

- Your user identification code (UIC) matches the job's UIC.
- Your group UIC matches the job's group UIC.
- SYSPRV privilege is enabled.
- BYPASS privilege is enabled.
- READALL privilege is enabled.
- You hold the READ, WRITE, or EXECUTE identifier for that job.

You have WRITE access to a job if at least one of the following is true:

- Your UIC matches the job's UIC.
- Your group UIC matches the job's group UIC and you have GROUP privilege enabled.
- SYSPRV privilege is enabled.
- BYPASS privilege is enabled.
- You hold the WRITE identifier for that job.

You have EXECUTE access to a job if at least one of the following is true:

- You have WRITE access to the job.
- You hold the EXECUTE identifier for the job.

To use the general commands of the manager on remote nodes of the network, you must have an authorized account on the remote node.

For a description of identifier-based access to jobs, see Identifier-Based Access to Jobs in *Chapter 3, "Getting Started with Job Management Manager"*.

1.5.1. Privileges for Management Commands

To use management commands of the manager, you must have one of the following OpenVMS system privileges:

- OPER
- SYSPRV

If you are unsure what your system privileges are, see your system manager.

Some management commands require other privileges. These privileges are described in the respective command descriptions.

For a description of general and management commands, see *Chapter 4, "Using the DCL Command Interface"*.

1.6. Wide Area Network Support

You can use Job Management Manager to control and coordinate the processing of jobs among nodes in a wide area network. Note the following requirements:

- The manager must be installed on both the sending and receiving nodes of the network.
- When you install the manager on each node, answer yes to the wide area network support question in the installation procedure.

The manager can be installed with support for executing jobs on remote OpenVMS nodes. If you want to use the manager to control jobs on one or more of these remote nodes, answer YES to this question. This feature of the manager requires that a supported TCP/IP stack is installed and running. For details on supported TCP/IP products and version, see <https://docs.vmssoftware.com/vsi-openvms-x86-64-v923-installation-guide/>.

The TCP/IP network startup procedure must execute before the manager startup procedure in the system startup procedure.

In a wide area network, users can:

- Enter commands for the manager on remote nodes, if they have either a proxy account or valid account and password to access the remote nodes.
- Add remote job dependencies if they have READ access to the remote jobs.

In a wide area network, system managers can:

- Set the interval and expiration time for attempting to resend job-dependency notifications between remote nodes when communication failures occur.
- Control the information that the manager enters in the event log on remote nodes.

For functions that involve timing between different nodes, such as maximum run time and start time for remote or batch mode jobs, system times must be synchronized.

The manager keeps a record of network information in two files:

NSCHED\$:RETRY.LOG

This file contains a history of all network operations that the manager has retried. You can type or print this file. The system manager can delete the file when it becomes too large.

NSCHED\$:RETRY.DAT

This file is present only when retry activity is pending.

The manager uses this file to control operations for resending mail notifications over the network. Deleting this file stops all retry attempts currently in progress. You cannot type or print the file. Instead, use the **SCHEDULE SHOW NETWORK** command to display its contents.

Remote job termination message contents are the same as regular termination messages except:

- Peak page is not used.
- BIO is total IOS.
- DIO is not used.
- Volumes are not used.

When you show a job locally, it always knows the correct exit status (success or failure) of remote mode jobs. When showing that same job from across the net, the mode can be unknown and the exit status tests for SUCCESS/FAILURE/UNKNOWN based on the status number (odd or even).

Changing the mode of a currently running job can cause the results of the current run to be improperly reported. The termination information is discarded if the job starts with one mode and exits with another. Changing can also cause some internal counters to not be properly updated, such as the number of currently running jobs. Always verify that a job is not currently running before changing its mode. Available modes are DETACHED, BATCH, and REMOTE.

Prefunctions and postfunctions are not supported for remote mode jobs, which run on remote agents. For remote OpenVMS jobs, prefunctions and postfunctions work as usual.

The following fields are not accessible remotely; to control them the user must log directly in to the remote account:

SJOB	EXID	QUEUE-NAME
TJOB	WRID	CPU-LIMIT
RDID	MODE	REMOTE-USER/NODE

Chapter 2. How Job Management Manager Works

2.1. Overview

This chapter provides general information about how Job Management Manager (the manager) works. The following sections provide specific information about how to access and use the manager.

2.1.1. Starting and Stopping Job Management Manager

To start the manager, at the DCL prompt, enter the following:

```
$ @sys$startup:job$manager$startup.com
```

To stop the manager, at the DCL prompt, enter the following:

```
$ @sys$startup:job$manager$shutdown.com
```

2.2. Automating and Managing Jobs

The primary goal of Job Management Manager is to automate and manage repetitive jobs. This section provides an overview of how the manager achieves this goal.

2.2.1. Job, Job Database, and Job Record

To understand how the manager manages jobs, you should understand how it defines a job, a job database, and a job record.

Job

The main part of a Job Management Manager job can be a single system-level command or a command file containing multiple commands. A job also includes all the related information required to process commands, such as when and where to run the commands and how to complete them. You refer to a job by its job name or job number.

Job Database

A job database is the collection of information that tells the manager when and how to process each particular job. The Job Management Manager database is a multikeyed index file that uses OpenVMS record management services (RMS) to store information about each job.

The manager is most useful when processing repetitive jobs, such as fiscal reporting procedures, system maintenance or backup tasks, and user applications. The job database is used to process repetitive jobs.

You can also use the manager to run non-repetitive jobs that would benefit from its scheduling, job synchronization, or other features.

Job Record

A Job Management Manager job is represented in the database by a job record. You create most of the job database information when you create, copy, or modify job characteristics, such as:

- Job name, group, type
- Primary command, prefunction, and postfunction
- Schedule interval and run times
- Job dependencies
- Message notification conditions
- Job restart conditions

The primary part of a job record is a Job Management Manager command, which is any valid system-level command you want to perform. To enter a command, you use the standard command format for the operating system that the command executes on. In the DCL interface, use quotes to enclose a case-sensitive command.

Any user can create a job record or copy a record to the Job Management Manager job database.

Note

The default location for files referenced in a Job Management Manager command is your SYS\$LOGIN device and directory. You can specify a different device and directory as part of your file specification.

The following table documents the limit for text attributes in the job record:

Job Attribute	Maximum Length
Cluster node	6
Comment	80
Group	40
Job name	40
Load balance group	14
Mail to	50
Output (job log file)	255
Queue (batch mode)	31
Postfunction	1024
Prefunction	1024
Remote node	72
Restart parameter	40
Type	40
Username	32
VMS command	1024
All time specifications	23

2.3. How Job Management Manager Processes Jobs

By default, the manager runs a job as a detached process on OpenVMS. The manager is activated by one of two actions:

- An internal timer starts the job at its scheduled starting time, except as restricted by dependencies, Special Days, or resource availability.
- The manager receives a message to look at a particular job. The message may be the result of a Job Management Manager command (such as Create, Modify, or Run), or a completed job.

By default, the manager does not use OpenVMS batch processing to run jobs. The manager creates the job process directly.

The process created by the manager first executes the system-wide and user-specific login command files. Then it executes the command file `SCHEDULER$SHELL.COM` supplied with the manager, in the `NSCHED$`: logical directory.

`SCHEDULER$SHELL.COM` runs a program that extracts the job's commands from the Job Management Manager database and performs them. When the process created by the manager exits, OpenVMS sends a completion message to the manager. The manager then updates its job database accordingly.

2.4. How Job Management Manager Reports a Completed Job

After performing the last command in a job, the manager reports the job as complete. If the job is a command procedure, the completion status of job is the status of the last command performed within the procedure.

If the last command in a job is successful, the manager reports a successful completion status. For example, the manager would always report a successful completion status for the following procedure, even if the **COPY** command failed. The reason is that the last command performed is `WRITE SYS$OUTPUT`.

```
$ ON ERROR GOTO ERR_EXIT:
$ COPY A.B C.D
$ EXIT
$ ERR_EXIT:
$ WRITE SYS$OUTPUT "file copy failed!"
$ EXIT
```

Because, the completion status can affect the execution of other jobs depending on this job; you can modify your command procedure to exit with the desired status.

To modify the preceding example to complete with the desired status, change the error-handling section as follows:

```
$ ERR_EXIT:
$ SAVED_STATUS = $STATUS
$ WRITE SYS$OUTPUT "file copy failed!"
```

```
$ EXIT SAVED_STATUS
```

By default, the manager treats failures of any level as job failures. For example, the error

%NSCHED-F-JOBSNOTFND is considered a FATAL error by OpenVMS instead of a less severe error. You can control this level by modifying the job status before it finishes; the postfunction is recommended as the place to do this modification. For example, a DCL procedure can be written to verify and change the NSCHED_STATUS symbol to achieve the desired job completion effect. Bits 0–2 of the NSCHED_STATUS can be selected for the severity level of the failure, as listed here. Alternately, the symbol NSCHED_SEVERITY can be tested, with the same result.

- 0 = Warning
- 1 = Success
- 2 = Error
- 3 = Informational
- 4 = Severe Error
- 5-7 = Reserved by VSI

The OpenVMS command STOP/ID, when issued for a Job Management Manager job process, causes the process to exit with the status of the last image that was executing. The command turns out to be SUCCESS more often than not, instead of an expected FAILURE or ERROR. The JOB\$MANAGER\$SHUTDOWN.COM procedure issues **SCHED STOP/REQUEUE** command which aborts all running jobs before stopping NSCHED process. These jobs are assigned %NSCHED-F-JOBABORT status.

2.5. Job States

Job Management Manager jobs are always in one of six system-assigned states:

- Scheduled
- On hold
- Job slot wait
- Dependency wait
- Running
- Queued

You can request the manager to change the state of a job by using one of the following commands:

- Abort
- Delete
- Hold
- Release

- Run

These commands set a requested state in the Job Management Manager database. The manager grants a request when it processes the related job's database record.

2.5.1. Restarting Jobs

When you create a job in the database, you can specify whether or not to restart the job after system interruptions. You can also include restart labels in the job's command procedures, if you want the job to restart from a specific step rather than at the beginning.

2.6. Job Database Listing

Job Management Manager updates job information each time it runs a job. That information includes:

- Unique job number (assigned when the job is created)
- Current job state
- Last run time and exit status
- Next scheduled run time
- Success and failure counts
- Last run-time error message (if any)
- A list of satisfied and unsatisfied dependencies

To see a complete job database listing for one or more jobs using the DCL interface, you use the **SCHEDULE SHOW JOBS/FULL** command. The following example displays a job database listing:

```
Job Name Entry User_name State Next Run Time
-----
SAMPLE_JOB 206 EXAMPLE Scheduled 17-APR-2003 03:00
VMS_Command : @DISK$USER1:[EXAMPLE]SAMPLE_JOB.COM
Group : (none) Type : (none)
Comment : This job runs nightly as a basic test
Last Start Time : 17-Apr-2003 03:00
Last Finish Time : 17-Apr-2003 03:10 Last Exit Status : SUCCESS
Schedule Interval : D 03:00 Mode : Detached
Mail to : EXAMPLE (No Mail)
Days : (MON,TUE,WED,THU,FRI)
Output File : DISK$USER1:[EXAMPLE]SAMPLE_JOB.LOG
Cluster_CPU : Default Notify user upon completion
Load Balance Group: GROUP1
Run Priority : Default
Max_Time Warning : None Job Always retained
Stall Notify : None No Retry on Error
Success Count : 166 Failure Count : 13
Owner UIC : [201,203] Restart on Crash
No Pre or Post Function for this job
No local jobs depend upon this job.
```

This job has no Dependencies on other jobs in the Motif interface, select which jobs to display by clicking on the **Load Jobs...** button in either the Graphical or Linear Display window.

2.6.1. Specifying a Job

Format

You can specify a Job Management Manager job by name or number. The job name or number is named the "job specifier".

```
( [node::] job-name [=username] )
```

Example:

```
XYZ:WEEKLY-RENAME=DOE
```

Default Node and Username

You do not have to specify a node or username for jobs that you own.

- The default node is the default agent on your local OpenVMS Cluster.
- The default username is the username for the account you are logged into.

Specifying Another User's Job

To specify a job that belongs to another user, use the *=username* qualifier to add the user's name to the job name. In the following example, the job COPY_IT belongs to user SMITH:

```
COPY_IT=SMITH
```

Specifying a Job Outside Your OpenVMS Cluster

You can specify Job Management Manager jobs on nodes outside your OpenVMS Cluster if the manager is installed with its wide area network features. For example, to specify the job COPY_IT on node XYZ outside your OpenVMS Cluster, add that node name to the job name.

```
XYZ::COPY_IT
```

Uniqueness of a Job Name and Number

- Job names are unique within a specific user's set of jobs. For example, if user Doe and user Smith each own jobs named COPY_IT, the manager treats them as two separate jobs.
- Job numbers are unique within an OpenVMS Cluster.

Note

The manager assigns each job in the database a unique job number. For this reason, VSI does not support the use of numerical job names.

Important

Never use the statement "READ SYS\$COMMAND <symbol>" in any step of a Job Management Manager job (pre, main, or post). During the execution of Job Management Manager jobs, SYS\$COMMAND points to the SCHEDULER\$SHELL command procedure. The effect of the

READ statement is to read a command from the SCHEDULER\$SHELL.COM file and assign it to the symbol. When the user command finishes, the next line in the SCHEDULER\$SHELL should be "EXIT \$STATUS." However, if this line has been read and assigned to the user symbol, the file pointer now points to the next line in SCHEDULER\$SHELL.COM and the **EXIT** command is skipped, causing the job run process to start again.

2.7. Job Output Files

Job output files are written to the default device of job owner and directory as specified in the authorization file for the account.

You cannot use logical names defined in the LNM\$PROCESS_TABLE to direct the location of the output file of the Job Management Manager job. The logical names that you use must be visible before you execute the user login procedure.

2.8. Using Wildcards

You can use wildcard characters when specifying jobs in Job Management Manager commands. Follow the standard OpenVMS wildcard conventions. For example:

- The asterisk (*) wildcard is a substitute for any character or string of characters in a job specification.
- The percent (%) wildcard is a substitute for one character only.

The following examples show how to use wildcards in a full Job Management Manager job specification.

Examples:

- This example shows a full job specification without wildcards:

```
XYZ : :WEEKLY-RENAME=DOE
```

- This example shows a full job specification for all jobs on node XYZ that have job names beginning with WEEKLY and are owned by users whose OpenVMS usernames begin with the letter D:

```
XYZ : :WEEKLY*=D*
```

- This example displays a list of the current user's jobs that begin with the letter W, on the local OpenVMS Cluster:

```
W*
```

Note

You cannot use wildcards in job specifiers when creating or modifying a job dependency list with the /SYNCHRONIZATION qualifier in the DCL interface.

2.9. Accessing Jobs on Remote Nodes

To perform an operation on a remote node, you must have authorized access to an account on that node. You also must have the appropriate privileges to perform the specified command.

There are three ways to access jobs on remote nodes:

- Specify your default proxy account on the remote node. You must have a default proxy account on that node. For example:

```
XYZ : :WEEKLY-RENAME
```

- Specify the remote node and a username for a particular proxy account in quotation marks. You must have a proxy account on that node. For example:

```
XYZ "SMITH" : :WEEKLY-RENAME=DOE
```

- Specify the remote node and an access control string within quotation marks. For example:

```
XYZ "DOE MYPASSWORD" : :WEEKLY-RENAME
```

2.10. Prefunctions and Postfunctions

A *prefunction* is a separate command performed *before* the job's primary command begins running. For example, a prefunction can include selecting for a necessary file or verifying that a tape is mounted before a backup procedure begins.

A *postfunction* is a separate command performed *after* the job's primary command has completed. For example, a postfunction can include printing a directory of files created by a **COPY** command, or purging duplicate versions of files created by the job's primary DCL command.

Multiple jobs use prefunctions and postfunctions to reduce programming effort and increase standardization.

In the DCL interface, you can add prefunctions and postfunctions by using the **/PRE_FUNCTION** and **/POST_FUNCTION** qualifiers with the following commands:

- **SCHEDULE CREATE**
- **SCHEDULE COPY**
- **SCHEDULE MODIFY**

For example:

```
$ SCHED MODIFY/PRE_FUNCTION="DIR/PRINT[DOE.PROJ-C]*.LJT;*"
$ SCHED MODIFY/POST_FUNCTION="@CHECK_DISC.COM "
```

In the DECwindows interfaces, you can enter prefunctions and postfunctions directly in the fields provided when creating, copying, or modifying a job.

2.10.1. Prefunction and Postfunction Processing Rules

- If the prefunction ends with an error, Job Management Manager does not process the primary command, and ends the entire job with an error.
- If the prefunction completes successfully, the manager processes the primary command of job.
- The manager processes the postfunction, regardless of whether the primary command of job completes successfully or terminates with an error.
- If the postfunction completes successfully, the manager notifies you that the job completed successfully. If the primary DCL command of job ended with an error, the manager reports those errors in the specified output file of job (if any).

- If the postfunction ends with an error, the manager notifies you that the entire job ended with an error, even if the primary DCL command of job completed successfully.

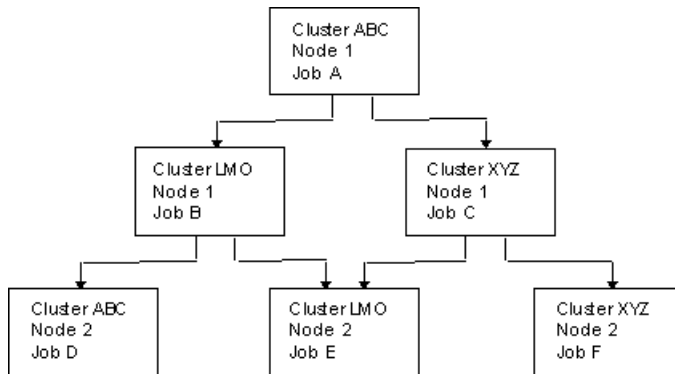
2.11. Job Dependencies

You can create a Job Management Manager job to run in any of the following three ways:

- Standalone (does not depend on other jobs or involve a prefunction or postfunction)
- Within the constraints of a prefunction and postfunction
- Synchronized to depend on other jobs

The manager allows a job to have an unlimited number of job dependencies. For example, a primary job can depend on up to 16 other Job Management Manager jobs; each of these jobs can depend on 16 more jobs, and so on. These jobs can be on the same node, within an OpenVMS Cluster, or on remote nodes of a wide area network.

The following diagram illustrates a job dependency hierarchy among several remote nodes of a network:



In the example:

- Job B and Job C **depend** on Job A. When Job A runs successfully, Job B and Job C can run.
- Job D **depends** on Job B. When Job B runs successfully, Job D can run.
- Job F **depends** on Job C. When Job C runs successfully, Job F can run.
- Job E depends on Jobs B and C. **Both** Job B and Job C must run successfully before Job E can run.

2.11.1. Remote Nodes

The wide area network feature allows you to:

- Use Job Management Manager commands on remote nodes, if you have either a *proxy account* or a valid account and a password to access those nodes.
- Add job dependencies on remote jobs, if you have READ access to those jobs. If you are setting up remote synchronizations to jobs that run on a cluster that has no cluster alias, you need to specify the dependency to one specific node in that cluster and then node restrict the job running on that cluster. Without the node restriction, the job may be load balanced to any node in the cluster and the dependency may not be satisfied.

When you set up remote synchronizations to jobs that run in a cluster that has an alias, always set the remote synchronization in terms of the cluster alias. This set up lets the dependency be satisfied regardless of what node the job runs on. If the job on the cluster is node restricted, set the dependency to that node.

The following commands and qualifiers are not supported over the network:

Commands	Qualifiers
COPY	/MODE /QUEUE /EXID /RDID /WRID /SJOB /TJOB /OPCOM /CPULIMIT /REMOTE_NODE /SYNC=remotenode::job /LOAD_BALANCE_GROUP
CREATE	/MODE /QUEUE /EXID /RDID /WRID /SJOB /TJOB /OPCOM /CPULIMIT /REMOTE_NODE /SD_CLASS /SYNC=remotenode::job /LOAD_BALANCE_GROUP
DELETE	/SD_CLASS /IGNORE
HELP	
MODIFY	/MODE /QUEUE /EXID /RDID /WRID /SJOB /TJOB /OPCOM /CPULIMIT /REMOTE_NODE /SD_CLASS /SYNC=remotenode::job /LOAD_BALANCE_GROUP
SCRIPT JOB	
SET JOB	
SET RESTART_VALUE	
SHOW DELETES	
SHOW DEPENDENCIES	
SHOW FISCAL_YEAR	
SHOW JOB	/PROCESS /QUOTAS /ACCOUNTING /RIGHTS /PRIVILEGES /VERIFY_REMOTE
SHOW SD_CLASS	

2.12. Delta Time Formats

Job Management Manager uses two different delta time formats: OpenVMS delta time and Job Management Manager delta time.

2.12.1. OpenVMS Delta Time

Some Job Management Manager commands use qualifiers that require you to specify time intervals using OpenVMS delta time. Here is the format for OpenVMS delta time:

`dddd-hh:mm:ss.cc`

- `dddd` – Days
- `hh` – Hours
- `mm` – Minutes
- `ss` – Seconds
- `cc` – Hundredths of a second

Note

At the DCL prompt, the **/MAX_TIME** and **/STALL_TIME** qualifiers require use of OpenVMS delta time; in all other interfaces, including the shell, these qualifiers use Job Management Manager delta time.

2.12.1.1. Rules for Formatting OpenVMS Delta Time

- You can omit fields within the time only if you include the punctuation that separates the fields. For example, `2::10` means 2 hours and 10 seconds.
- You can truncate a delta time on both the right and the left. In the example `2::10`, the days and hundredths-of-a-second fields are truncated.

2.12.2. Job Management Manager Delta Time

The **/INTERVAL** qualifier, used with the **COPY**, **CREATE**, and **MODIFY** commands, allows you to use Job Management Manager delta time. Here is the format for Job Management Manager delta time:

`+dddd-hh:mm:ss.cc`

- `+dddd` – Days
- `hh` – Hours
- `mm` – Minutes
- `ss` – Seconds
- `cc` – Hundredths of a second

Note

Job Management Manager delta time (`+dddd hh:mm:ss.cc`) is NOT the same as OpenVMS delta time (`dddd-hh:mm:ss.cc`).

2.12.2.1. Rules for Formatting Job Management Manager Delta Time

- You can omit fields within the time only if you include the punctuation that separates the fields. For example, `"+0 2::10"` means 2 hours and 10 seconds.
- You can truncate a delta time on the right. In the example `"+0 2::10,"` the hundredths-of-a-second fields are truncated.

- You must specify a value for days (+dddd), even if it is 0.
- In the DCL interface, you must enclose the Job Management Manager delta time in quotes because it always contains a space after the day value.

2.12.2.2. Time Creep in Jobs

When a job's schedule interval is a delta time expression, for instance, "+00 15:00," the manager calculates the job's next scheduled run time by adding the time interval to the job's actual start time. Typically, there is a small time lapse (usually a second or two) between a job's scheduled start time and the time it actually begins to run. When delta time schedule intervals are used, this time delay is reflected in the job's next scheduled run time. For example, a job that was scheduled to start at 11:30:00.00 may be next scheduled to run at 11:45:01.35, reflecting a "time creep" of 1.35 seconds.

You can use the **/USE_NEXT_TIME** qualifier when creating or modifying a job that uses a delta time interval. This will calculate the next scheduled run time from the *scheduled* time, not from the actual start time, preventing time creep. However, if a user uses the RUN command to run the job manually, the job will add the delta time interval to the next scheduled run time. This problem can be avoided by using the command **RUN/NORESCHEDULE** when running the job manually.

Alternatively, you can create several similar jobs with specific schedule intervals. For example, to run a job every fifteen minutes exactly (beginning on the hour, 15 minutes past, 30 minutes past, and 45 minutes past), create four similar jobs with the following schedule intervals:

- H 00:00
- H 15:00
- H 30:00
- H 45:00

When calculating a job's Next Scheduled Run Time, the manager starts by finding the start of the next period specified in the Schedule interval. For example, if using a schedule interval of "D 22:00," the manager first finds the start of the next Day and then applies the hour specification. The resulting Next Scheduled Run Time always lies within the next period of time. For example, if a job with an interval of "D 22:00" is created (or run) at 10AM today, the next scheduled run time for that job will be tomorrow at 22:00 (Not today at 22:00). This behavior is expected.

Note

You must keep the system times for all nodes in a cluster synchronized. Time differences between cluster nodes can result in job dependencies not being processed correctly or jobs running repeatedly.

2.13. Special Days

Special Days are those days requiring particular consideration when scheduling jobs. You can prevent a job from running on Special Days or restrict a job to run only on

Special Days. Examples of Special Days are New Year's Day, Thanksgiving, and Christmas, and company holidays, plant shutdown days, and vacation days.

Special Day Classes are collections of calendar dates, grouped for specific reasons (holidays, plant closing days, and so on).

Job Management Manager Special Day support lets you:

- Define Special Day Classes.
- Define as many Special Days as you like within each class.
- Control how jobs are scheduled on Special Days.
- Use combination of Special Day Classes to affect the scheduling of any job.

You need OPER or SYSPRV permission to create, modify, or delete a Special Day Class.

In the DCL interface, elements of a Special Day Class are defined in terms of absolute date specifications, such as 24-MAR-2012. In the DECwindows interface, you define Special Days by clicking days in a calendar.

See Defining Special Day Classes, for Special Day class naming rules.

Once defined, one or more Special Day Classes can be used in combination as a scheduling constraint for a job, with either of the following controls:

- Restricting the job from running on the Special Days specified
- Requiring the job to run only on the Special Days specified

2.13.1. How Special Days Affect Run Times

Any time a job runs either automatically or as the result of a RUN command, the next scheduled start time for the job is calculated. This calculation takes into consideration the job's scheduling interval, day restrictions, and Special Day Restrictions.

If a Special Day conflict is detected, requiring that the job be rescheduled or put on hold, any messages to be issued to the job's owner and the operator will be sent at this time. Since this calculation is performed at the time a job is run, messages about a job's Special Day scheduling conflicts may be sent before the run of the job has completed.

This is normal and expected behavior.

2.13.1.1. Dynamic Changes to Special Day Classes

Because Special Day Restrictions are taken into account only when calculating a job's next scheduled start time, and a job's next scheduled start time is only calculated when a job is run, any changes made to a job's Special Day Restrictions will not take effect until the next time a job is run and its next scheduled start time is calculated. Thus, creating Special Day Restrictions for a job, adding or deleting Special Day Classes from a job's Special Day Restrictions, or modifying the contents of a Special Day Class will not change a job's current scheduled run time. These changes will take effect the next time the job is run and its next start time is calculated.

Remote Sites

Special Day Restrictions work the same way for all jobs, whether they run locally, remotely, or have dependencies.

Note

The **/SD_ACTION** qualifier is not supported when a server is specified.

For a job running on a remote site, the Special Days relevant to that remote site may be defined on the scheduling node and used as a Special Day Restriction for the remote job. For example, if a job runs on a machine in Japan, but its schedule is determined by the manager running on a machine in the United States, the Special Days relevant to Japan must be defined on the machine in the United States. They may then be used as Special Day Restrictions in determining the remote job's next scheduled run time.

For jobs in a dependency tree, Special Day Restrictions are taken into account only when determining the job's next scheduled run time. The job will run once its next scheduled run time has passed, and its dependencies are satisfied.

Note

Special Day Restrictions on a job can also affect the run time of another job that has a dependency on that job.

Special Days are not taken into account under the following conditions:

- When the first start time for a job is calculated as it is created
- When you use the **SCHEDULE SHOW SCHEDULE** command

2.13.2. Creating the First Special Day Class

When you create the first Special Day Class in the database, the manager creates the file `NSCHED$:SCHED$SD_CLASSES.DAT` with the file protection `S:WRED,O:WRED,G:RE,W:RE`. If you later copy or back up this file with a different file protection, problems may occur when you use the Special Day Classes.

2.13.3. /SD_ACTION Qualifier

The **/SD_ACTION** qualifier is valid only for jobs that have the **/NOT_ON** qualifier specified for them. When a job has the **/ONLY_ON** qualifier specified, Special Day actions (HOLD and SKIP) have no effect.

2.14. Using Batch Mode

Job Management Manager can use *batch* as an execution engine instead of using *detached* jobs. A job must be specially created or modified to use batch mode. The default mode is DETACHED.

2.14.1. Job Management Manager Batch Characteristics

Job Management Manager batch jobs have the following unique characteristics:

Characteristic	Description
QUEUE	The name of a submit queue. The queue must exist on the system, and defaults to <code>SYSS\$BATCH</code> .
QPRIORITY	Priority to submit the job into the queue. The default is 100, minimum value is 1, and maximum value is 255.
CPULIMIT	The number of 100ms ticks of CPU that can be used before the job is terminated by the system. The default is 0 = infinite. The minimum is 50.

Characteristic	Description
ENTRY NUMBER	The entry number of the job in the queue. Only present when the job is in queue.

2.14.2. Job Management Manager Batch Execution

The manager queues the job at the specified next run time for the job. The job starts executing when the queues on the system allow it to execute.

If the queues are stopped, full, stalled, or in some other nonrun state, the job remains marked as Queued until the queue allows the job to execute. In this respect, the jobs are controlled by the system batch queues.

If the queue is generic, the job may run on a different node than that specified.

When a job's mode is set, all the mode-specific characteristics are reset to the default values, or to specified values at the time of setting the mode. These values can be modified later.

Any characteristics of the queue will impact Job Management Manager jobs, including the queue's run priority.

Note

QUEUE, QPRIORITY, and CPULIMIT are supported only for batch mode. Only detached mode supports NODE_RESTRICTION and RUN_PRIORITY. Only remote mode supports REMOTE_NODE in the form of either *user@node* or *node*.

If a job's execution queue is on a machine that does not have the manager installed, the job does not execute properly, and is marked as a failed run attempt.

Note

Resource accounting information for Job Management Manager batch jobs in the event log has slightly smaller values than the corresponding process-id information in the OpenVMS accounting file. Also, peak pagefile use and total mounted volumes are not available.

If a job is deleted while pending, the manager will not actually delete the job before execution.

During the time a job is in a queued state, if a request for another run of the same job comes in, the existing request is deleted from the queue and the new one replaces it. Such requests will thus be processed until the job reaches the head of the queue and starts its run.

During the time a job is in a queued state, if a request for characteristic modifications comes in, the existing request is deleted from the queue and the new one replaces it. Such requests will thus be processed until the job reaches the head of the queue and starts its run.

Note

If SYS\$COMMAND is used in a command procedure being run in the Job Management Manager batch mode (instead of to OpenVMS batch mode without the manager), it may run the command file twice.

2.14.3. Batch Mode Logicals

The following logical names supported by the manager provide flexibility around Batch job support:

SCHED\$BATCH_KEEP_LOG

If TRUE, the log files created by Job Management Manager jobs run in BATCH queues are kept. Otherwise, these log files are deleted when the job completes. This logical is defined initially in JOB\$MANAGER\$STARTUP.COM. The default value for this logical is TRUE.

SCHED\$BATCH_NAME_PREFIX

You can use this logical to override batch name prefix of Job Management Manager. If you define this logical, the value of it is used as the prefix to the batch job name. If you do not define this logical, the default value VJM is used. This logical is not defined by JOB\$MANAGER\$STARTUP.COM.

SCHED\$BATCH_PRINT_LOG

If TRUE, the log files created by Job Management Manager jobs run in BATCH queues are printed to SYS%PRINT. Otherwise, the log files are not printed. This logical is defined initially in JOB\$MANAGER\$STARTUP.COM. The default value for this logical is FALSE.

2.15. Job Management Manager in an OpenVMS Cluster

You can run Job Management Manager jobs only on a node where the manager is running. This rule also applies to OpenVMS Cluster nodes.

To see if the manager is running on your node or OpenVMS Cluster, enter the DCL command **SCHEDULE SHOW STATUS**. To see if a Job Management Agent is running on a remote node, use the **/REMOTE_NODE=remote_node** qualifier.

On an OpenVMS Cluster, you can enable a job to run:

- On only one particular node
- On any node
- On a subset of nodes

You can specify a node for a job's execution by using the **/NODE (/CLUSTER_NODE)** or **/REMOTE_NODE** qualifier when creating, copying, or modifying a job.

You can force a job to run manually on a particular node regardless of any node restriction in the job database. To do this, specify the **/NODE (or /CLUSTER_NODE)** qualifier with the **SCHEDULE RUN** command.

You can specify a subset of cluster nodes for a job's execution by using the **/LOAD_BALANCE_GROUP** qualifier when creating, copying, or modifying a job.

The maximum number of nodes that the manager can support in an OpenVMS Cluster is 200. This feature of the manager is designed to accommodate any future OpenVMS Cluster support increases up to 200 nodes.

The manager is supported on homogeneous and mixed-mode OpenVMS clusters. You should have only one active SYSUAF.DAT file in an OpenVMS Cluster. Use of more than one SYSUAF.DAT file within an OpenVMS Cluster is not supported by VSI (see the *OpenVMS Cluster Systems Manual*).

2.15.1. How the Default Node Runs a Job Management Manager Job

When you run the manager on multiple nodes of an OpenVMS Cluster, one node becomes the *default executor*. If load balancing is turned off, this node runs all jobs that do not have a node restriction. If load balancing is turned on and the logical name NSCHED\$LBAL\$WAIT_AVAIL_NODE is not defined (default) or is set to FALSE, a job that cannot be scheduled to any node in its Load Balance group will run on the default node.

Normally, the first node in the OpenVMS Cluster that starts the manager becomes the default executor; however, the system manager can control this by using the **SCHEDULE SET DEFAULT** command.

If the default agent node fails or the manager is stopped on that node, another node takes over this function.

To see which node is the current default agent; enter the **SCHEDULE SHOW DEFAULT** command.

2.15.2. Supporting Batch Mode Jobs in an OpenVMS Cluster

To support batch mode jobs in an OpenVMS Cluster, you should make all batch queues accessible to all nodes in the cluster. If this is not desirable, set up all queues to be accessible to the default node.

If you use the **SHOW JOB** command from a node that cannot access the batch queue for the job, the command will not be able to find the job.

2.15.3. Balancing Job Loads Among Nodes in an OpenVMS Cluster

Job Management Manager provides the opportunity to use dynamic load balancing in an OpenVMS Cluster environment, and runs jobs on the computer that has the lowest current system activity. Your system manager can enable or disable this activity.

Load balancing is enabled by default for all systems in an OpenVMS cluster. To enable load balancing, use the **SCHEDULE SET LOAD_BALANCE** command. When you enable this feature, Job Management Manager scans all computers in the cluster that are currently running the manager and selects the least loaded of these nodes. The manager assigns the job to run on that node. You can also restrict a job to run on a subset of nodes in a cluster using Load Balance Groups. If you assign a Load Balance Group to a job, the job runs on the least loaded of the nodes in the Load Balance Group. All jobs are load balanced unless you turn off load balancing or a job is restricted to a particular node.

To verify if load balancing is currently enabled, use the **SCHEDULE SHOW LOAD_BALANCE** command.

For a description of how load balancing works, see *Chapter 3, "Getting Started with Job Management Manager"*.

The manager uses a currently least-loaded node approach to determine which node to run a job on. For more information about the algorithm the manager uses, see *Chapter 3, "Getting Started with Job*

Management Manager". If you must customize further, you can invoke a command procedure to use a different method to assign jobs to nodes in a cluster. For more information about how to specify your own load balancing algorithm, see the appendix "Customizing Load Balancing."

2.15.4. Recovering Jobs from Nodes That Fail

If a node fails while a job is running, you may want to restart the job on another node. Or, you may decide to put the job on hold and notify the job's owner to perform manual cleanup.

You can control whether or not to restart a job by specifying the **/RESTART** or **/NORESTART** qualifier with the DCL commands **SCHEDULE CREATE**, **SCHEDULE MODIFY**, or **SCHEDULE COPY**, or by choosing Restart options in the DECwindows interface.

You can also restart a job from a particular checkpoint in the job. To set up checkpoints, use the **SCHEDULE SET RESTART_VALUE** command in your DCL command procedure.

If a node on which the manager is running fails, and the manager is running on at least one other node in the OpenVMS Cluster, the manager:

- Updates the status of any interrupted jobs to aborted.
- Notifies users of the failure.
- Restarts the job, if another CPU is available and if the job has been set with the **/RESTART** qualifier.

The manager evaluates the error messages it receives and determines whether the failure is due to its inability to create a detached process or to the job itself failing. If a job has **/RESTART** set and fails because of a system error that prevents creation of an OpenVMS process, the job is rescheduled according to its interval; **/RETRY** has no impact on when the job is rescheduled. If the job was created with the **/NORESTART** qualifier, the job is put on hold and is not restarted automatically.

Chapter 3. Getting Started with Job Management Manager

3.1. Job Management Manager Overview

Job Management Manager (the manager) is a powerful and flexible tool. When you use the manager, you can choose from a variety of options and customize the manager to fit the way you do business.

This chapter covers the following topics and points you to additional information:

- Roles and responsibilities
- User interfaces
- Steps leading toward automation
- Controlling job access
- The callable interface (SDK)
- System management and normal maintenance
- Job Management Manager/Agent performance tuning
- Load balancing

3.2. Roles and Responsibilities

With Job Management Manager, Operations people must perform fewer repetitive chores manually. They must understand their new roles and responsibilities.

Decide at the outset the person or persons who must the following:

- Convert existing procedures to run under the manager. They must understand the business processes, dependency networks, and the jobs that are going into the system.
- Oversee operations until new procedures run smoothly.
- Monitor operations after the system is set up and running correctly.

These roles can be full-time jobs, depending on the complexity of your production schedules.

3.3. User Interfaces

The manager's flexibility allows you to choose from two user interfaces. The following table can help you select the interface to use:

Use this interface...	If you need to...
DCL	Initiate and monitor operations using general and system management commands.

Use this interface...	If you need to...
	Create dependencies.
DECwindows Motif	Create, modify, and monitor jobs as a general user. Create dependencies for complicated job streams as a general user. Obtain a graphical view of Job Management Manager jobs.

3.4. Steps Leading Toward Automation

A major benefit of Job Management Manager is its ability to automate time-consuming and labor-intensive tasks. The first steps toward automation include:

- Understanding your own production schedules, dependency networks, and what must be automated
- Asking yourself what would happen if:
 - A job is run by mistake or out of sequence
 - A job does not run when you want it to
 - System time is set incorrectly
 - A dependency is removed from a job by mistake
- Devising a naming convention to let you use wildcards on groups of jobs across accounts
- Dividing jobs into steps so they can take advantage of FAILOVER/RESTART at a step
- Collecting commonly used functions into prefunctions and postfunctions that can be shared among jobs
- Modifying existing procedures to remove extraneous logic, such as for rescheduling, job dependency, or error notification
- Selecting error handling and job exiting status
- Using job dependencies and load balancing for "parallel processing"
- Putting some simple, non-critical jobs into the job database to become familiar with how the manager works
- Devising a logical grouping scheme that takes advantage of the "GROUP" and "TYPE" capabilities of manager, over and above the wildcard capability

3.5. Controlling Job Access

Job Management Manager provides flexibility for security, allowing you to grant specific job access by specific users to specific jobs.

There are two major ways to control job access:

- UIC privileges

- Identifiers

For a description of UIC privileges, see the Chapter "An Overview of Job Management Manager."

3.5.1. Identifier-Based Access to Jobs

In addition to the UIC privilege-based security scheme, you can control job access through Job Management Manager using rights identifiers. Each job may have the following identifiers:

- READ
- WRITE
- EXECUTE

If the requesting process also holds that identifier, then access is granted. This feature allows access control to individual jobs on a user-by-user basis, and eliminates the need for privileges for operations personnel. It also removes the need for privileged proxy accounts for remote control of privileged jobs.

These identifiers...	Also permit these accesses...
WRITE	EXECUTE and READ
EXECUTE	READ

The identifiers that are supported in the manager are "general identifiers," or identifiers that can be created with the OpenVMS **AUTHORIZE** utility and stored in the file **RIGHTSLIST.DAT**:

- **ADD/IDENTIFIER**: **OpenVMS AUTHORIZE** command to create an identifier
- **GRANT/IDENTIFIER**: **OpenVMS AUTHORIZE** command to grant an identifier to a user

Identifier qualifiers do not appear in the DCL user interface as prompts to the **CREATE**, **MODIFY**, or **COPY** commands; they must be explicitly specified on the command line. In the DCL interface, the following qualifiers control the identifiers:

- **/RDID**, for READ access
- **/EXID**, for EXECUTE access
- **/WRID**, for WRITE access

Identifier qualifiers appear in the DECwindows user interface. The following privileges are required to set job identifiers:

- WRITE access to the job is required to set the READ identifier.
- SYSPRV privilege is required to set the EXECUTE identifier.
- CMKRNK privilege is required to set the WRITE identifier.

Only one identifier value is supported for each of the identifier access qualifiers.

3.5.2. Mixed Environments

In mixed computing environments, users without system privileges can have security problems initiating and controlling jobs.

One solution is to fill in the following information sheet, use it to create a command procedure, and request a person with SYSPRV privileges to run the command procedure.

Job Management Manager—Job Scheduling Information Sheet

Use this sheet to help track your job scheduling Information:

*** = optional**

Job Command Information

Main Function to Run:

* Prefunction to Run:

*Postfunction to Run:

Job Information

Job Owner Name:

Job Name:

*Group Name:

*Type Name:

*Node to Run Job on:

*Remote Node to Run Job on:

Detached/Batch/Remote Mode to Run Job:

Retain Job upon Success/Failure/All/None:

*Cluster Nodes to restrict job to LOAD_BALANCE_GROUP:

Schedule Information

Start Time:

Interval:

Days of Week:

*Special Day Classes:

Notification Information

Send Mail on Success/Failure/All/None:

*Send Mail on Special Day Action:

*Mail Addresses:

Terminal Broadcast Message:

OPCOM Message:

Error Recovery Information

Restart on Crash:

Retry on Job Failure:

*Retry Interval:

*Retry Quantity:

*Stall Notify

*Job to Run When Stalled (SJOB):

*Max Time Notify:

*Job to Run When Max Time Exceeded (TJOB):

Logging Information

*Output Logfile Name:

Job Access Security Information

*Read Access Identifier (RDID):

*Execute Access Identifier (EXID):

*Write Access Identifier (WRID):

Other Job Run Information

*Dependent on What Other Jobs (SYNCHRONIZATION):

*Run Priority:

*Comment

Batch Queue Execution Specific Information

Queue to Run Job:

*Queue Priority (QPRIORITY):

*CPU Limit for Job:

3.6. The Callable Interface (SDK)

VSI Job Management for OpenVMS supports a callable Software Developer's kit (SDK) on OpenVMS. This interface lets you write programs that access VSI Job Management for OpenVMS. During installation, the library files NSCHED_SUBS.OLB and CLIB.OLB are placed in the NSCHED\$EXE

directory. These files contain libraries of subroutines and C utilities that you can call to access VSI Job Management for OpenVMS features and functions from within the program.

In addition to the API library, the following files are placed in the NSCHED\$DATA directory when you install the callable interface option:

- VSSDEFS.H
- HCIDEF.H
- HCISELECT.H
- NSCHED_JOB_FIELDS.H
- EXAMPLE_CREATE.C

For more information about the callable interface, see the *VSI Job Management for OpenVMS Programming Guide*.

3.7. System Management and Normal Maintenance

This section contains some hints from experienced Job Management Manager users.

System management is simplified with the manager. However, Job Management Manager detached processes behave differently from commands running under OpenVMS batch mode. Comparing Job Management Manager Detached Processes with OpenVMS Standard Batch Processes in the "An Overview of Job Management Manager" chapter gives a detailed comparison between the manager and OpenVMS Batch Processing.

Because privileged operators can modify or delete jobs, privileges can be used with caution. As an alternative, you can consider using EXECUTE identifiers, to allow efficient running without privileges.

You can create user-defined keys for frequently used Job Management Manager DCL commands.

At a minimum, operators must know how to do the following:

- Tell what jobs are running. For example:

```
$ SCHEDULE SHOW JOBS/STATUS=RUNNING
```
- Abort a job
- Hold or release groups of jobs (in case a problem arises)
- Set or clear dependencies for jobs to resume the production cycle

The following hints can help you perform normal maintenance:

- If the log file grows beyond 5000 blocks, it slows product performance. Use the following command to close the log file:

```
SCHEDULE CLOSE LOG_FILE
```

This command renames the file to filename.OLD and creates new version of the file.

- To include summary information in the new file, use this command:

```
SCHEDULE CLOSE LOG_FILE/SUMMARIZE
```

- You can create a Job Management Manager job to execute this command at regular intervals. You could also write a procedure like the following and run it daily:

```
$ SCHED_LOG = "NSCHED$:VERMONT_CREAMERY.LOG"
$!
$ IF F$FILE(SCHED_LOG,"EOF") .GT. 2000
$ THEN
$ SCHEDULE CLOSE LOG_FILE
$ MAIL NL: SYSTEM/SUBJECT="THE SCHEDULER LOG FILE HAS BEEN CLOSED"
$ ENDIF
$ EXIT
```

- If many jobs are added and deleted, the job database becomes fragmented. Use the following command to show the number of deleted jobs:

```
$ SCHEDULE SHOW DELETES
```

- If the number of deleted jobs is greater than 250-300, stop the manager, and then use the following command to run a program that defragments your job database:

```
$ SCHEDULE OPTIMIZE DATABASE
```

Note

SCHEDULE OPTIMIZE DATABASE resets the counter of deleted jobs to zero. You must restart the manager after running **OPTIMIZE DATABASE**.

You can stop the manager while jobs are running as long as it starts again before they complete.

If a job completes while the manager is stopped, the job finishes typically, but the manager gets a false status--"job was aborted". If the job is set up to restart, it does a job restart.

Use the command **SCHED SHOW SCHEDULED_RUNS** to predict future runs of specified jobs. This command uses the historical job run data to predict the job run schedule, and provides an output qualifier that allows you to print a hardcopy report of the schedule. See the *VSI Job Management for OpenVMS Command Reference Guide* for additional information and associated qualifiers for this command.

Use the command **SCHED SHOW HISTORY** for information about the execution history of selected jobs. The history log name can be specified; by default the current log is used to extract the requested information. See the *VSI Job Management for OpenVMS Command Reference Guide* for additional information and associated qualifiers for this command.

3.7.1. Dependent Jobs Relying on a Single Condition

You can use a postfunction to create a conditional situation between two jobs. For example, after Job A completes, and if a single condition is met, you want Job B to execute.

If you want a dependent job to run only after one specific ending condition from the main function, use a simple DCL procedure similar to the following:

```
$ IF NSCHED_STATUS .EQ. the-condition-you-want
```

```
$ THEN
$ EXIT 1
$ ELSE
$ EXIT 0
$ ENDIF
```

3.7.2. Synchronizing Local and Remote Databases

For functions that involve timing between different nodes, such as maximum run time and start time for remote or batch mode jobs, system times must be synchronized.

If you are concerned about synchronizing local and remote databases, there are at least two areas of synchronization to consider:

- Time synchronization

For time synchronization, simply verify the time of each server initially. The manager does not do any time correction for remote sites. Remote job dependencies occur because the remote dependency completed, and sent a RUN message across the net to the local dependent (or vice versa), *not* because of any time synchronization.

- Job dependency synchronization

For job dependency synchronization, verify the following:

- The proxy for remote account is set up for the local account.
- The reference in the **SYNCHRONIZATION** qualifier for remote servers is correct.

The manager does the rest of it.

3.7.3. Activating the Debug Log File

Activate the debug log file if you are having a problem, and need to track messages, actions, and times.

Follow the steps in the table below to activate the debug log the next time the manager starts. If you want to turn on debugging immediately, use the **SCHED SET DEBUG ON** command.

Step	Action
1	Edit the startup procedure SYS\$STARTUP:JOB\$MANAGER\$STARTUP.COM.
2	Locate the line that tells the manager to start up in debug mode, and make sure this line is no longer commented out.
3	Restart the manager.

Turning debugging on creates a log file that describes messages, actions, and times. One log file is created in NSCHED\$DATA for each machine running the NSCHED process. The file is called 'nodename'.LOG.

In most cases the log file is fairly small. However, the log file that belongs to the default manager could get as large as 5000 blocks. To free disk space, simply purge the log file. The manager creates a new log file at each startup, and purges the log files to limit them to two versions.

You need not create log files until you have a reason to use them. In general, if you have plenty of free space, write the log files.

3.8. Load Balancing

Load balancing depends on the system's *rating*.

3.8.1. Rating a System

The system rating is the average between the *Memory_rating* and the *CPU_rating*. When jobs are ready for execution, they find the node with the highest rating, and send the job to that node.

The algorithm for calculating ratings follows:

```
Total Rating = Cpu_rating + Memory_rating
```

```
Cpu_rating = Cpu_calculation * CPU_weight *  
(1 + ((number of CPUs - 1) * .2)) / number of CPUs
```

```
Memory_rating = Memory_calculation * Memory_weight
```

```
Cpu_calculation = ( %Free CPU * (VUPS per CPU) * #CPUs  
* COM-FACTOR ) + (3 * (CPUs* VUPS per CPU))
```

```
COM-FACTOR = 0.5 if the ((number of COM+CUR+COMO processes on the  
system)-1) is more than the number of CPUs
```

```
COM-FACTOR = 1.0 if the ((number of COM+CUR+COMO processes on the  
system)-1) is equal to the number of CPUs
```

```
COM-FACTOR = 1.5 if the ((number of COM+CUR+COMO processes on the  
system)-1) is less than the number of CPUs
```

```
Memory_calculation= (FREE_PAGES/TOTAL_PAGES) * MEG * 10
```

If Current Jobs is greater than or equal to **MAX_JOBS**, then rating = 0

CPU_weight and Memory_weight have values 0 through 1.

To view the "VUPS per CPU" rating of the system, use the following command:

\$SCHEDULE SHOW CPU_RATING

This command displays the CPU (VUPS) rating of the system used in the algorithm's calculation. For example, the command returns the following for a DS20E:

```
$SCHEDULE SHOW CPU_RATING  
Machine Name <XYZ> Hardware Name <COMPAQ AlphaServer DS20E 5000 MH>  
Hardware Model <1921> Rating <1000>
```

If the system does not have a CPU rating in the Job Management Manager CPU database, Job Management Manager returns a message saying the CPU is not rated. If you see this message, contact VSI Technical Support and provide them with the text of the message. The following is a sample message for an unrated machine:

```
$SCHEDULE SHOW CPU_RATING
```

```
Machine Name <XYZ> Hardware Name <COMPAQ AlphaServer DS20E 5000 MH>  
Hardware Model <1921> is not rated
```

When jobs are loaded, the manager identifies the node with the highest rating and loads the first job on that node. Then the manager checks to see which node has the highest rating and loads the next job onto that node. Each time the manager determines the node with the highest rating, it checks the node's MAX_JOBS value (the maximum number of jobs allowed) before loading the job. Jobs are loaded on each node until that node's MAX_JOBS value is reached.

A node's rating is updated:

- Every NSCHED\$LBAL\$INTERVAL seconds
- Every time a job starts on that node
- Every time a job completes on that node

You can adjust the CPU rating in the following way. Define the logical name NSCHED\$LBAL_INSENSITIVE to a non-zero integer percentage value. When this logical is greater than zero, the load balancing algorithm considers all systems that are rated within the given percent of each other to be the same rating. They are then load balanced using round robin scheduling. This distributes the load more equitably among similarly rated systems. We recommend you define this logical name cluster wide, or in the system table of each node in the cluster.

You can adjust when a rating is updated in the following way. Define the logical name NSCHED\$LBAL\$UPDATE_RATING to ON. The manager will decrease a node's rating when a job is sent to it, so a node is aware of the number of jobs pending and the rating routine will take these jobs into consideration. We recommend you define this logical name cluster wide, or in the system table of each node in the cluster.

Note

VSI does not recommend having an NSCHED\$LBAL\$INTERVAL interval of calculation below 5 seconds. The resources used to calculate the ratings become high, and the return on these resources is low.

The rating is calculated **only** on the node where the job state changes. If you want to use load balancing on a particular node, you will get an accurate rating for it **after** the job is started, or after its regular update.

There is no minimum rating a node must have in order to have a job started on it. To remove a node from load balancing, set the rating weights for that node to 0. If the rating is 0, the node is not used in load balancing. When a node has reached its MAX_JOBS value, its rating is zero.

3.8.2. How Load Balance Groups Work

You can balance a job load further by specifying groups of nodes in a cluster that certain jobs may run on. The creation and deletion of load balance groups is a management function. To specify these groups, follow these steps:

1. Define a Load Balance Group using the following command:

```
$SCHEDULE CONFIG CREATE LOAD_BALANCE_GROUP
```

2. Add one or more nodes to the group using the following command:

```
$SCHEDULE CONFIG ADD NODE node_name /LOAD_BALANCE_GROUP=group_name
```

3. Add the Load Balance Group name to the job definitions using the following command:

```
$SCHEDULE CREATE /LOAD_BALANCE_GROUP
```

or

```
$SCHEDULE MODIFY /LOAD_BALANCE_GROUP
```

Note

A Load Balance Group name can have 14 alphanumeric characters. The name can also include an underscore "_".

Scheduling Notes

The following notes provide further information about scheduling load balancing and Load Balance Groups. These scenarios assume the manager is installed on all nodes in a cluster.

Default Behavior, Cluster

By default, load balancing is enabled for all systems in a cluster. If no Load Balance Group is specified for a job on a cluster, the job is scheduled to all nodes in a cluster.

Default behavior, standalone

By default, load balancing is disabled on non-clustered systems. Load Balance Group information for a job is ignored.

No Load Balance Group nodes available

When a job is scheduled to run but nodes are not available in the Load Balance Group, the job's behavior is based on the setting of the logical NSCHED\$LBAL\$WAIT_AVAIL_NODE. If this system logical is TRUE, the job moves to the Job Slot Wait state until a node is available in the Load Balance Group. If this system logical is FALSE, the jobs run on the default node in the cluster.

Empty Load Balance Group

A Load Balance Group definition may be empty. In this case, all nodes in the cluster are Unselected, so the job does not run on any machine. If the system logical NSHED\$LBAL\$WAIT_AVAIL_NODE is TRUE, the job moves to the Job Slot Wait state until a node is available in the Load Balance Group. If this system logical is FALSE, the jobs run on the default node in the cluster.

Deleted Load Balance Group

Deleting a Load Balance Group removes any reference to that group by a job definition. Those jobs begin to schedule to all nodes in the cluster. If you do not want this scheduling, do not delete a Load Balance Group as a first step. You can modify the jobs to assign them to another Load Balance Group or remove all nodes from the Load Balance Group. You can use the following command to help understand the impact of deleting the Load Balance Group:

```
SCHEDULE SHOW JOB */LOAD_BALANCE_GROUP=group-name-to-be-deleted
```

This command shows you all jobs associated with the Load Balance Group.

Group specification override

The job cluster_node field, if not empty, overrides the group specification. This job runs on the specified node, regardless of the Load Balance group assigned to the job, whether the node is a member of that Load Balance group or not.

Batch and Remote Mode

A job defined with a mode of Batch or Remote is not eligible for Load Balancing. The selection of a load balance group in the Auxiliary Job Definition window is disabled. The Group Editor button remains active.

Special Case Job and Load Balance Group Definitions

The following are special cases with Job and Load Balance Group definitions:

A node with group definitions and jobs referencing these groups is moved from a cluster to a standalone system:

In the Motif interface, you can create Load Balance group definitions; however, only the local node is available to add to a Load Balance group. If a node is removed from a group, it is temporarily added to the list of Unselected Nodes. However, when the job modification is complete, it is deleted from the available list. In the Configuration CLI interface, any node can be added or removed from a group (see **SCHEDULE CONFIG ADD NODE**). For scheduling, load balancing is disabled by default on a standalone node. A job can only run on the standalone node.

A node with group definitions and jobs referencing these groups is moved from a cluster to another cluster:

A job referencing a group with nonexistent nodes does not load balance to nodes that do not exist in the current cluster. In the Motif interface, group definitions can be created and deleted. However, only cluster nodes from the new cluster are available to add to any group. If a node from the former cluster is removed from a group, it is temporarily added to the list of Unselected Nodes. However when the job modification is complete, it is deleted from the available list. In the Configuration CLI interface, any node can be added or removed from a group (see **SCHEDULE CONFIG ADD NODE**). For more information, see the section Scheduling Notes.

A standalone node with no group definitions:

In the Motif interface, you can create group definitions; however, only the local node is available to add to any group. In the Configuration CLI interface, you can add or remove any node from a group (see **SCHEDULE CONFIG ADD NODE**). Load balancing is disabled by default on a standalone node. A job can only run on the standalone node.

A cluster node with no group definitions:

In the Motif interface, group definitions can be created using nodes available in the cluster to add to any group. In the Configuration CLI interface, any node can be added or removed from a group (see **SCHEDULE CONFIG ADD NODE**). Load balancing is enabled by default. For more information, see the section Scheduling Notes.

3.9. Job Management Manager/Agent Performance Tuning

To accommodate the diversity of environmental conditions that may exist at different installation sites, Job Management Manager can tune the communication code that lets the manager and the agent communicate.

Sites with slow networks, unreliable networks, heavily burdened nodes, or heavy spikes in CPU utilization may need to adjust these parameters. Tuning parameters have the same names for the manager and for the agent.

These parameters may be logical names or symbols. If the parameters are system logical names, they affect the NSCHED, SCHED\$LISTENER retry, and user interface processes. If you want different parameters for these processes, use symbols or process logical names created in the process's startup file.

The following table lists the parameters with their possible settings:

Parameter	Default	Minimum	Maximum
SCHED_PACKET_WAIT	2 sec	1 sec	60 sec
SCHED_MAX_RETRIES	3	0	32
SCHED_PENDING_DELAY	60 sec	1 sec	–

SCHED_PACKET_WAIT and SCHED_MAX_RETRIES work together. When one component (manager or agent) tries to communicate with the other, the component tries to connect. If it fails to connect, the component waits the amount of seconds set in SCHED_PACKET_WAIT and tries again. The component repeats this process a number of times equal to SCHED_MAX_RETRIES.

Note

Since packet communication is synchronous, the sending process waits a maximum of $\text{SCHED_PACKET_WAIT} * \text{SCHED_MAX_RETRIES}$ seconds before ending attempts to send a packet. If you set both parameters high, the component may wait an unacceptably long time before starting other work. A system that has many jobs starting or finishing within a short time of one another requires higher values for these parameters.

Chapter 4. Using the DCL Command Interface

4.1. Overview

This chapter describes how to use Job Management Manager (the manager) with DCL commands. The chapter covers the following topics:

- Accessing the manager and entering commands
- General and management commands
- Creating a Job Management Manager job
- Creating job dependencies
- Assigning a job to a group or type
- Special days
- Sending OPCOM messages
- Creating a job in batch mode
- Summary of Job Management Manager commands

4.2. Accessing Job Management Manager and Entering Commands

Job Management Manager commands are divided into general commands for users and management commands for system managers. All Job Management Manager commands are available at the DCL command level, if you have privileges to use the command.

You can use the following ways to enter commands at the DCL command level:

From the DCL prompt (\$)

All general Job Management Manager commands begin with the word **SCHEDULE**. Example:

```
$ SCHEDULE SHOW JOBS
```

From the **SCHEDULE>** prompt

If you enter the command **SCHEDULE** at the DCL prompt and press Return, you invoke the manager as a subsystem. The **SCHEDULE>** prompt is displayed. You can enter Job Management Manager commands without the word **SCHEDULE**. Example:

```
$ SCHEDULE SCHEDULE> SHOW JOBS
```

To return to the DCL prompt, use the **EXIT** command or press Ctrl+Z.

From the **SCHEDULE Config>** prompt

Enter the **SCHEDULE Config** command for the subset of commands used to configure Load Balance Group definitions:

```
$ SCHEDULE CONFIG  
SCHEDULE Config> CREATE LOAD_BALANCE_GROUP GROUP1
```

To exit the **SCHEDULE Config>** prompt, type **EXIT** twice.

You can abbreviate the commands according to standard DCL abbreviation rules. If an abbreviation is unrecognized or ambiguous, the system displays an error message. If you omit a required parameter to a command, the manager prompts for it.

Recall of commands is supported via standard DCL rules.

4.2.1. Is Job Management Manager Installed?

To check if the manager is installed on your system and to see which nodes of an OpenVMS Cluster currently have the manager running, enter the following command:

```
$ SCHEDULE SHOW STATUS
```

If the manager is not installed, contact your system manager for more information.

To see whether a remote agent is installed on a remote node running another operating system, use the following command:

```
$ SCHEDULE SHOW STATUS/REMOTE_NODE=nodename
```

4.2.2. Using Help

This document describes all Job Management Manager commands. You can also get help online by entering either of the following commands:

```
$ HELP SCHEDULE  
$ SCHEDULE HELP
```

4.3. General and Management Commands

Job Management Manager commands are divided into two groups, for users and system managers. For the system privileges required to use each group of commands, see Privileges Needed to Use Job Management Manager in the chapter "An Overview of Job Management Manager."

4.3.1. General Command

General commands are for Job Management Manager users. They allow you to:

- Create or copy a new job into the Job Management Manager database.
- Run, abort, delete, or hold a job.
- Display information about a job.
- Modify an existing job.

- Override a job's normal characteristics for one specific run.
- Add a job to the dependency list of another job. You must have READ access to the dependency job.

For a summary of commands, see the Summary of Job Management Manager Commands. For a description of individual commands, see *VSI Job Management for OpenVMS Command Reference Guide*.

4.3.2. Management Command

Management commands are for system managers. They allow you to set up, monitor, and manage the Job Management Manager environment, on an OpenVMS Cluster or among remote nodes on your network. For example, management commands let you set the following:

- The default node for running the manager on an OpenVMS Cluster, for routine or failover job activity
- The default priority of Job Management Manager jobs
- The Job Management Manager load-balancing feature on a local OpenVMS Cluster
- The maximum number of Job Management Manager jobs that can run on all nodes or on a specific node on the OpenVMS Cluster
- The time interval and expiration time for resending job-dependency notifications between remote nodes, when a network communication failure occurs
- The type of information put in the event log

Management commands also let you:

- Close and rename the event log
- Delete pending attempts to resend job-dependency notifications on remote nodes
- Run the event-log reporting program
- Stop the manager on an OpenVMS Cluster, on a specific node of an OpenVMS Cluster, or on a specific remote node

Most management commands begin with the word **SCHEDULE**. However, some management commands are standard DCL commands that perform a specific task in support of Job Management Manager.

For a summary of commands, see the Summary of Job Management Manager Commands. For a description of individual commands, see *VSI Job Management for OpenVMS Command Reference Guide*.

4.4. Creating a Job Management Manager Job

To create a Job Management Manager job, use the **SCHEDULE CREATE** command. When you create the job, you can assign a job name and specify other information about the job such as:

- Output file
- Specific node on which to run the job
- Schedule interval

- Days of the week on which the job may run

The following is an example of how to create a job with the **SCHEDULE CREATE** command. When you use the **/PROMPT** qualifier with the command, the manager prompts you for all the information you can specify about that job.

```
SCHEDULE> CREATE/PROMPT
_VMS Command: @SYS$DEVICE:[EXAMPLE.PROJECT-C]WEEKLY-COPY.COM
_Job Name [None] : WEEKLY-RENAME
_Group [None] : WEEKLY
_Type [None] : PROJECT-COSTS
_Stall Notify [None] :
_Schedule Interval [None] : F W6 18:00:00
_Days [ALL]: (NONE,FRI)
_Special Day Restrictions [NONE]:
_Start Time [ 7-DEC-2011 18:00:00.04] :
_Output file [None] : SYS$DEVICE:[EXAMPLE.PROJECT-C]WEEKLY-
                        RENAME.REP
_Mode (Detached, Batch, Remote [Detached] :)
_Cluster Node [Default] :
_Retain (Success, Error, None or All [ALL]:) ALL
_Restart (Yes, No [Yes]:) YES
_Retry (Yes, No [No]:) YES
_Notify (Yes, No [Yes]:) NO
_Send-Mail (All, Error, Success or None [NONE]:) ALL
_Send MAIL on Special Day Action (Yes, No [No]:)
_Use_next_time (Yes, No [No]:) YES _OPCOM (Yes, No [No] :)
_Mail_Address [EXAMPLE] : EXAMPLE
_User_Name [EXAMPLE] : EXAMPLE
_Run Priority [Default] :
_Pre Function [None] : DIR/PRINT[EXAMPLE.PROJECT-C]*.REP
_Post Function [None] : DIR/PRINT[EXAMPLE.PROJECT-C.WEEKLY]*.REP _
_Maximum Time Warning [None] :
_Maximum Time Job [None] :
\_-_LBAL Group [None] :
_Synchronization [None] :
_Comment [None] : Check for weekly rep's/copy to subdirectory and rename/
check new rep files
%NSCHED-I-RQSTSUCCSS, Job 10521 - Created
```

4.5. Creating Job Dependencies

You can specify job dependencies for a job when you create, copy, or modify a Job Management Manager job. Job dependencies are other jobs that must complete successfully before the current job can run. You can specify up to 16 job dependencies for a job.

To specify job dependencies, use the **/SYNCHRONIZATION** qualifier with the **SCHEDULE CREATE**, **SCHEDULE COPY**, or **SCHEDULE MODIFY** command. You can specify a job dependency by its job name or job number in the Job Management Manager job database. To specify more than one job dependency, enclose the list in parentheses and separate the dependencies with commas.

Example:

```
SCHEDULE> MODIFY COPY_IT /SYNCH=(BACKUP,XYZ::RENAME_IT=DOE,XYZ::6)
```

This command modifies a job named COPY_IT to depend on three other jobs:

- Your own job `BACKUP`, in your local OpenVMS Cluster
- A job `RENAME_IT`, owned by user `DOE` in the OpenVMS Cluster that contains node `XYZ`
- Job 6 in the OpenVMS Cluster that contains node `XYZ`

4.5.1. Job-Dependency Restrictions

The following restrictions apply to creating or modifying job dependencies:

- If you have `READ` access to jobs belonging to other users, you can add these jobs to your job's dependency list by including the `=username` specifier with the job name or by specifying the job number.

Example:

```
/SYNCHRONIZATION= (BACKUP=SYSTEM, BACKUP=DOE, BACKUP=SMITH)
/SYNCHRONIZATION= (57=SYSTEM, 101=DOE, 566=SMITH)
```

- You cannot use wildcard characters in job specifiers when creating or modifying a job dependency list with the `/SYNCHRONIZATION` qualifier.

4.5.2. Job-Dependency Processing Rule

Before a job can run, all jobs on which it depends must run successfully, at a time later than the dependent job's synchronization time. The *synchronization time* is either the last time the job's dependency list was modified, or the last time the job completed — whichever occurred most recently.

For example, if Job C depends on Jobs A and B, then Job C will not run until Jobs A and B have successfully run following Job C's synchronization time. If Job D is added to Job C's dependency list, then Jobs A and B, as well as D must run again before Job C can run.

This is because Job C's synchronization time would reflect the time that Job D was added to its dependency list.

4.5.2.1. Overriding Job Dependencies

You can override one or more job dependencies for a single job run by using the following command:

```
SCHEDULE> SET JOB job-specifier -
/_SCHEDULE> /NO_DEPENDENCY=( dependency [, ...] ) )
```

This command tells the manager to treat the specified job dependencies job as having been satisfied for the next run of the job.

4.6. Assigning a Job to a Group or Type

You can assign a Job Management Manager job to a specific group or type. This feature lets you perform job actions on an entire job group or type, using a single **Job Management Manager** command. You can create or delete groups and types using the DCL interface only.

A group name or type name is a text string of up to 40 characters. For example, you might use `DAILY`, `WEEKLY`, or `MONTHLY` as a group name. You might use `PROGRAM`, `DOCUMENTATION`, `BACKUP`, or `PROJECT-COSTS` as a type name.

Examples:

- This command inserts a job into a group named BACKUP:

```
SCHEDULE> MODIFY job-name/NEW_GROUP=BACKUP
```

- This command removes a job from a group:

```
SCHEDULE> MODIFY jobname/NONEW_GROUP
```

- This command shows all your jobs that begin with the letter A in group WEEKLY:

```
SCHEDULE> SHOW JOB A*/GROUP=WEEKLY
```

- This command shows all jobs in the system whose type is BACKUP:

```
SCHEDULE> SHOW JOB */TYPE=BACKUP/ALL
```

- This command shows all jobs belonging to user DOE that are in any group beginning with WEEKLY and any type beginning with BACKUP:

```
SCHEDULE> SHOW JOB *=DOE/GROUP=WEEKLY*/TYPE=BACKUP*
```

4.7. Special Days

For a description of Special Day support, which allows you to schedule jobs to run only on certain days or not to run on those days, see Special Days in the chapter "How Job Management Manager Works." Special Days can be used to restrict jobs from running on company holidays, plant shutdown days, and so forth, or to run jobs only on certain collections of days. This is sometimes referred to as "holiday support."

4.7.1. Defining Special Day Classes

You can define Special Day Classes by creating a named class and then associating a list of date-expressions with it. Special Day Classes are governed by the following rules:

- Special Day Class names are restricted to 40 characters in length
- Class names cannot begin with a number
- They can contain only alphanumeric characters and the underscore character (`_`)
- The list of date-expressions can consist of one or more absolute date specifications
- You can specify the list of date-expressions associated with a Special Day Class at the time the Special Day Class is created, or add to it later
- You can modify the contents of a Special Day Class after its creation by inserting and removing elements

Acceptable Dates for Special Day Classes

Special Day Classes can contain dates beginning the year 1970 and continuing through the year 2037.

Required Privileges

Only users with OPER or SYSPRV privileges are allowed to create, modify, and delete Special Day Classes.

How to Define Special Day Classes

The following example shows how a Special Day Class is defined in the DCL interface:

```
SCHEDULE> CREATE MY_CLASS /SD_CLASS -  
_SCHEDULE> /DATES=(1-JAN-2012,25-DEC-2011,26-NOV-2011)
```

How to Add Days To Existing Special Day Classes

The following example shows how to add days to an existing Special Day Class.

```
SCHEDULE> MODIFY/SD_CLASS/DATES=(1-JAN,TOMORROW) classname
```

How to Remove Existing Days from a Class

The following example shows how to remove existing days from a Special Day Class.

```
SCHEDULE> MODIFY/SD_CLASS/NODATES=(TODAY,12-DEC-2011) classname
```

4.7.2. Restriction Expressions: Relating Special Days to Jobs

Once you define Special Day Classes, you can use one or more classes in combination as scheduling constraints for jobs. Defining Special Day constraints for jobs is supported in all the user interfaces of the manager.

You can provide scheduling restraints for jobs by associating a Special Day Restriction Expression with a job-specifier. A Special Day Restriction Expression is a list of one or more Special Day Classes that defines one of the following:

- Days on which a job should not run
- Days on which a job is restricted to running

The job-specifier can be a specific job name or use wildcards in the job name. The job-specifier can include a **GROUP** or **TYPE** indicator. For example:

```
SCHEDULE> MODIFY JOBX/GROUP=BATCH -  
_SCHEDULE> /SD_RESTRICTION=(COMPANY_HOLIDAYS,PLANT_SHUTDOWN_DAYS) -  
_SCHEDULE> /NOT_ON
```

Restricting a Job from Running on Special Days

You can restrict a job from running on certain days by using the **/NOT_ON** qualifier in conjunction with the job's schedule interval. For example:

```
SCHEDULE> MODIFY JOBX /SD_RESTRICTION=(COMPANY_HOLIDAYS)/NOT_ON
```

This example instructs the manager not to run JOBX on company holidays.

The manager calculates the job's next scheduled run time according to the job's schedule interval and then checks against its Special Day Restrictions. If the job's next scheduled run time falls on a restricted Special Day, the manager applies one of two user-specified Special Day Actions:

- Puts the job on hold

- Skips the scheduled run of the job and applies the schedule interval again to calculate a new next scheduled run time. This is done repeatedly until a day that satisfies the job's Special Day Restrictions is found.

In both cases, if the job has OPCOM or NOTIFY set, the owner of the job and the operator are notified of the Special Day conflict and what Special Day Action was applied.

Restricting a Job to Run Only on Special Days

You can restrict a job to run only on Special Days by using the **/ONLY_ON** qualifier. For example:

```
SCHEDULE> MODIFY JOBX/SD_RESTRICTION=(LAST_DAY_OF_FISCAL_QUARTER) /ONLY_ON
```

This example tells the manager to run JOBX *only* on the days in the class LAST_DAY_OF_FISCAL_QUARTER.

When the manager calculates the job's next scheduled run time, the job's schedule interval is applied. The manager then checks to see if the resulting day is a day the job is permitted to run. If not, the manager applies the schedule interval again until it finds a day the job is permitted to run.

If the manager does not find a day on which the job is permitted to run, it puts the job on hold. In either case, if the job has OPCOM or NOTIFY set, the owner and operator are notified of the conflict, and what Special Day Action was applied.

Restriction Expressions Are Mutually Exclusive

You can use either **/NOT_ON** or **/ONLY_ON** for any given job. You cannot use both Special Day qualifiers for the same job.

4.7.3. Special Day Actions

Special Day Actions specify what action the manager should take when a job's scheduled run time happens to fall on a day that is part of a Special Day Class. Special Day Actions are specified on a per-job basis. For example, users can instruct the manager to:

- Skip backups on restricted Special Days
- Put payroll programs on hold to be rescheduled by the operator

Setting a Job's Special Day Action

Following is an example of how to set a job's Special Day Action:

```
SCHEDULE> MODIFY MYJOB/SD_ACTION=HOLD
```

SD_ACTION can be HOLD or SKIP.

No Special Day Action Specified

If you do not explicitly specify a Special Day Action for a job, the manager applies the system default action at the time the job's Special Day Restrictions are created. For example, if you define the system default Special Day Action to be SKIP, then new Special Day Restrictions created for jobs have the default Special Day action of SKIP. If you redefine the system default Special Day Action to HOLD, new Special Day Restrictions have the default Special Day Action of HOLD.

You can redefine the system default by defining the logical NSCHED_DEFAULT_SD_ACTION. For example:

```
$ DEFINE NSCHED_DEFAULT_SD_ACTION HOLD
```

NSCHED_DEFAULT_SD_ACTION can be HOLD or SKIP. The default as shipped is SKIP.

Redefining the system default Special Day Action does not affect the Special Day Action of jobs with existing Special Day Restrictions.

4.7.4. Showing Information Related to Special Days

You can obtain information about the Special Days defined on your system by using the SHOW command. For example:

```
SCHEDULE> SHOW SD_CLASS name
```

You can show the Special Day Classes defined on the system, or see the definition of a specific Special Day Class.

When using the **SHOW** command, Special Day names can contain wildcards.

You can also use the **SHOW** command to find out what Special Day Restrictions are associated with a job. Job names can also use wildcards, and can include the **/GROUP** and **/TYPE** indicators. For example:

```
SCHEDULE> SHOW JOB/FULL
SCHEDULE> SHOW JOB/GROUP=BATCH/FULL
```

4.8. Sending OPCOM Messages

The Operator Communication Manager is a process for controlling communication between a process and a system operator. The manager lets you automatically send an OPCOM message upon job completion. This feature is part of the notification facility. You can send an automatic OPCOM message from the DCL User Interface by invoking the qualifier **/OPCOM**. (You can also specify that OPCOM messages be sent in the DECwindows Auxiliary Options dialog box.)

The OPCOM class that the manager uses for notification is the CLUSTER OPERATOR class. The operator messages are sent to an OPC\$B_MS_TARGET of OPC\$M_NM_CLUSTER.

For example:

```
SCHEDULE> MODIFY job-name/OPCOM
SCHEDULE> RUN job-name
```

This command will result in automatic notification when the job is complete.

4.9. Creating a Job in Batch Mode

The manager can use batch as an execution engine instead of using detached jobs. Create or modify a job specially to use batch mode. The default is **/MODE=DETACHED**.

The following examples show how to create a job in batch mode:

```
SCHEDULE> CREATE/MODE=BATCH/QUEUE=SYS$BATCH
```

This example creates a batch job without using the **/PROMPT** qualifier, and prompts only for a standard subset of qualifiers. See the *VSI Job Management for OpenVMS Command Reference Guide* for more details on **/PROMPT**.

```
SCHEDULE> CREATE/PROMPT/MODE=BATCH/QUEUE=SYS$BATCH
```

In this example the user creates a batch job using the **/PROMPT** qualifier, and is prompted for a set of qualifiers.

The **SCHED SHOW** command does not display attributes that are not allowable for current MODE of that job.

4.10. Summary of Job Management Manager Commands

The following two sections briefly describe all the commands in the manager. For complete descriptions and examples, see *VSI Job Management for OpenVMS Command Reference Guide*. If you enter commands at the DCL prompt (\$), remember to begin each command with the word **SCHEDULE**.

4.10.1. General Commands

Command	Description
ABORT	Stops one or more running jobs.
COPY	Creates a new job, copying the settings for an existing job.
CREATE	Creates a new job or a Special Day Class.
DELETE	Deletes one or more jobs or a Special Day Class from the database. This is a management command when used to delete a Special Day Class.
HELP	Displays the Help screens for the DCL interface.
HOLD	Puts one or more jobs on hold.
MODIFY	Modifies one or more existing jobs or a Special Day Class.
RELEASE	Takes one or more jobs off hold.
RUN	Runs the specified jobs, regardless of conditions such as job dependencies.
SCRIPT JOB	Creates a command procedure capable of recreating the specified job or jobs.
SET JOB	Lets you change the number and frequency of attempts to run a job. You can also override or reset job dependencies for a job. Can be used with EXECUTE privilege. (Requires WRITE for DELETE).
SET RESTART_VALUE	Specifies at what point in a job to restart the job, if a system problem occurs. See also SET JOB /RESTART_VALUE in the <i>VSI Job Management for OpenVMS Command Reference Guide</i> .
SHOW CPU_RATING	Displays the CPU (VUPS) rating of the system.
SHOW DEFAULT	Shows the default Job Management Manager node for a local or remote OpenVMS Cluster.

Command	Description
SHOW DELETES	Shows the number of jobs deleted from the job database since its last compression.
SHOW DEPENDENCIES	Creates a text file that represents a picture of a job's dependency structure.
SHOW FISCAL_YEAR	Displays the fiscal calendar.
SHOW HISTORY	Displays information about the execution histories of jobs in a given log file.
SHOW JOBS	Displays information about the specified jobs.
SHOW JOBS/PROCESS	Displays process-related information for currently running local jobs.
SHOW JOBS/SYMBOLS	Creates a symbol for each job characteristic for the specified job.
SHOW LOAD_BALANCE	Shows whether the load-balancing feature for a local or remote OpenVMS Cluster is on or off.
SHOW LOGGING	Shows the current events being logged for a specific node, all nodes in an OpenVMS Cluster, or remote nodes of a network.
SHOW MAX_JOBS	Shows the maximum number of jobs that can run simultaneously for a specific node, all nodes in an OpenVMS Cluster, or remote nodes of a network.
SHOW NAMES	Displays all the group and/or type names that a user currently has assigned for jobs.
SHOW PRIORITY	Shows the default run-time priority setting for a specific node, all nodes in an OpenVMS Cluster, or remote nodes of a network.
SHOW SCHEDULED_RUNS	Predicts future execution schedules for jobs based on their historical run data.
SHOW SD_CLASSES	Displays the contents of requested classes.
SHOW STATUS	Shows information about the manager operating for a specific node, all nodes in an OpenVMS Cluster, or remote nodes of a network.

4.10.2. Management Commands

Command	Description
CHECK	Tells a scheduler to do a consistency check on a database and update its internal structures and counters to reflect the database and the current state of the system.
CLOSE LOG_FILE	Closes and renames the event-log file while the manager is running.
CONFIG	Creates, modifies, or deletes Load Balance Group definitions.
DELETE	Deletes one or more jobs or a Special Day Class from the database. This is a management command only when used to delete a Special Day Class.
DELETE NSCHED \$:RETRY.DAT;	Deletes (stops) all remote node notifications currently being retried over the network. This command does not begin with SCHEDULE.
OPTIMIZE DATABASE	Reclaims unused space and optimizes the jobs database.

Command	Description
RUN NSCHED \$:DB_UTILITY.EXE	Repairs the jobs database. This command does not begin with SCHEDULE.
RUN NSCHED \$:VSS_REPORTS.EXE	Invokes the event-log report utility to select event-log report criteria and display or print a report. This command does not begin with SCHEDULE.
SET DEBUG	Sets debug logging for the manager.
SET DEFAULT	Sets the default manager node for a cluster.
SET FISCAL_YEAR	Defines a fiscal calendar to be used by the manager.
SET LOAD_BALANCE	Turns the load-balancing feature on or off in an OpenVMS Cluster.
SET LOGGING	Controls the event classes written to the Job Management Manager event-log file.
SET MAX_JOBS	Sets the limit for the maximum number of jobs that can be run simultaneously.
SET NETWORK EXPIRATION_RETRY	Sets the maximum time that a job-dependency notification on a remote node can be retried over the network.
SET NETWORK INTERVAL_RETRY	Sets the interval for retry of notification of job completion on remote nodes during communication failures on a wide area network.
SET PRIORITY	Sets default run-time priority for jobs.
SHOW NETWORK	Shows the retry interval, retry expiration time, retry alias, and information about retry operations in progress on the local OpenVMS Cluster.
STOP	Stops the manager on one or more nodes.
TYPE NSCHED \$:RETRY.LOG	Shows a history of every notification retry event attempted by a remote node. This command does not begin with SCHEDULE.

Chapter 5. Using the Motif Interface

5.1. Overview

This chapter describes how to use Job Management Manager (the manager) with the Motif interface. This interface allows you to use the mouse to click general user commands from menus and dialog boxes; the manager's management commands are not available in the Motif interface.

Before you begin using the Motif interface, you must be familiar with the basic DECwindows concepts described in the *Using DECwindows Motif for OpenVMS Guide*.

The first part of this chapter gives you an overview of the Motif interface:

- Using mouse buttons
- Using menus and buttons
- Starting the manager with the Motif interface
- **Job Management Manager** windows and symbols

The second part of the chapter describes how to perform major tasks in the manager:

- Loading jobs to display
- Clearing and changing the currently displayed jobs
- Adding to the currently displayed jobs
- Locating jobs by scrolling
- Viewing job information
- Creating a job
- Creating a job dependency
- Deleting a job dependency
- Controlling a job: aborting, holding, running, and releasing
- Editing a job
- Deleting a job
- Sorting jobs in the **Linear Display** window
- Creating a graphics file of your display

- Using navigation windows
- Special day classes
- Load Balance Groups
- Exiting from the Motif interface

The Motif interface uses a resource file to obtain values for several items. The resource file, named `SCHEDULER$MOTIF.DAT`, must reside in the `DECW$SYSTEM_DEFAULTS` or `DECW$USER_DEFAULTS` area.

During the installation, the following resource files are placed in the `NSCHED$` directory:

- `SCHEDULER$MOTIF_COLOR.DAT` — Sample resource file for color systems
- `SCHEDULER$MOTIF_BW.DAT` — Sample resource file for monochrome (single-plane) systems

Users on monochrome or gray-scale systems **MUST** copy the file `NSCHED$:SCHEDULER$MOTIF_BW.DAT` to their `DECW$USER_DEFAULTS` directory, and name it `SCHEDULER$MOTIF.DAT`. Unless copying `DAT` file is done, some aspects of the interface cannot display properly. Because the color version of the resource file is placed in the `DECW$SYSTEM_DEFAULTS` directory under the name `SCHEDULER$MOTIF.DAT`, users on color systems do not need to copy it to their `DECW$USER_DEFAULTS` directory unless they want to modify the default resource values.

Note

For DECwindows Motif, if you set your monitor to a resolution less than 1024 x 864, you can have to reduce font size to prevent clipping.

The `DECW$USER_DEFAULTS` area is defined to be `SYS$LOGIN` by default. Copying `NSCHED$:SCHEDULER$MOTIF_COLOR.DAT` to `DECW$USER_DEFAULTS` and naming it `SCHEDULER$MOTIF.DAT` allows you to change the resource file to meet your preferences without affecting the operation of the interface for other users.

The manager uses the Motif user interface. Old resource files (`SCHEDULER$XUI*.DAT`) are not used. To prevent those files from interfering with new Motif files, remove those files from the `NSCHED$` directory, from `DECW$USER_DEFAULTS`, and from `DECW$SYSTEM_DEFAULTS`. Also issue the following command and delete those executables to prevent them from interfering:

```
DECW$SYSTEM_DEFAULTS
```

5.1.1. Changing the Resource Files

To change the resource files, follow these steps:

1. Use the following command to copy the resource file to your DECwindows defaults directory:

```
$ COPY DECW$SYSTEM_DEFAULTS:SCHEDULER$MOTIF.DAT -  
_ $ DECW$USERDEFAULTS:*. *
```

By default, the logical `DECW$USER_DEFAULTS` points to `SYS$LOGIN`.

2. Open and edit the SCHEDULER\$MOTIF.DAT file to modify Job Management Manager resources. The file SCHEDULER\$MOTIF.DAT has comments describing different parameters that you can change.
3. Test the changes by restarting the DECwindows Motif interface. Repeat this step until the results match what you want.

5.1.2. Motif Interface

The Motif interface for the manager includes three main windows:

- **Graphical Display** window—Displays jobs as boxes that show you the current state of job and the result of its last run. Arrows indicate job dependencies, providing a process flowchart.
- **Linear Display** window—Displays jobs as a list you can sort by job number, job name, or job owner.
- **Real Time Message Display** window—Displays a log of all manager activity that takes place on the system during your Job Management Manager session.

5.2. Getting Started with the Motif Interface

Before you use the Motif interface, become familiar with the functions of mouse buttons, menus, and screen buttons.

5.2.1. Using Mouse Buttons

For the manager, mouse buttons perform the following functions:

This mouse button	Lets you...
MB1	• Pull down menus from the menu bar Choose menu items • Activate buttons by clicking on them • Create job dependencies • Select and unselect a job, making it the current object that a menu item acts on
MB2 on a two-button mouse	Display a pop-up menu for job actions, or for deleting a job dependency
MB3	Display a pop-up menu for job actions, or for deleting a job dependency

Menus are not available in the Real Time Message Display window.

5.2.2. Using Menus and Buttons

The following tables describe the uses of menus and buttons in the Job Display windows:

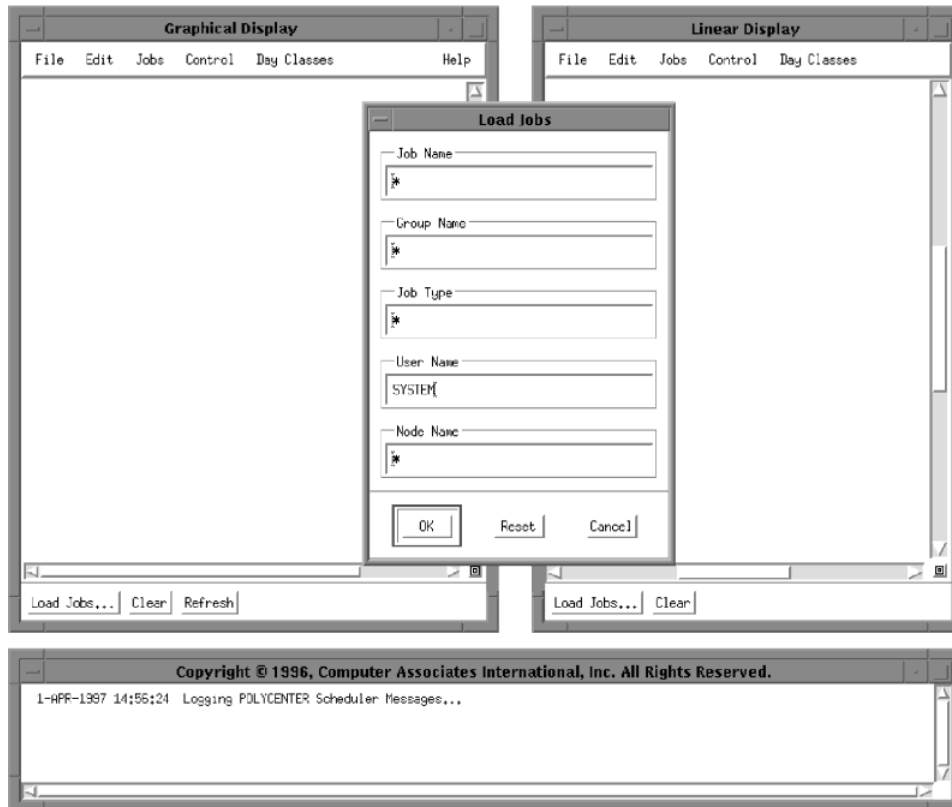
Menu	Uses
File (Linear Display window)	Quit from Job Management Manager

Menu	Uses
File (Graphical Display window)	● Quit from Job Management Manager ● Create a graphics file of the window contents
Edit (Graphical Display window)	● Load jobs Clear window ● Open Load Balance Group Editor
Edit (Linear Display window)	● Load jobs Clear window ● Open Load Balance Group Editor ● Sort jobs by job number, job name, or job owner
Jobs	● Create a job Edit a job Delete a job ● Display job output ● Monitor a job
Control	● Abort a job ● Put a job on hold Run a job ● Release a job that is being held
Day Classes	● Create Special Day Classes ● Edit Special Day Classes ● Delete Special Day Classes
Button	Use
Load Jobs...	Displays the Load Jobs dialog box (see The Load Jobs Dialog Box section). After you fill in the selection fields and click on OK, the interface fetches the appropriate jobs from the database and adds them to any jobs already displayed in the Jobs windows.
Clear	Clears all jobs from the Jobs window and removes them from memory.
Refresh	Redraws the contents of the Graphical Display window.

5.2.3. Starting Job Management Manager with the Motif Interface

To start the manager with the Motif interface, enter the following command at the DCL prompt:

```
$ SCHEDULE/INTERFACE=DECWINDOWS
```



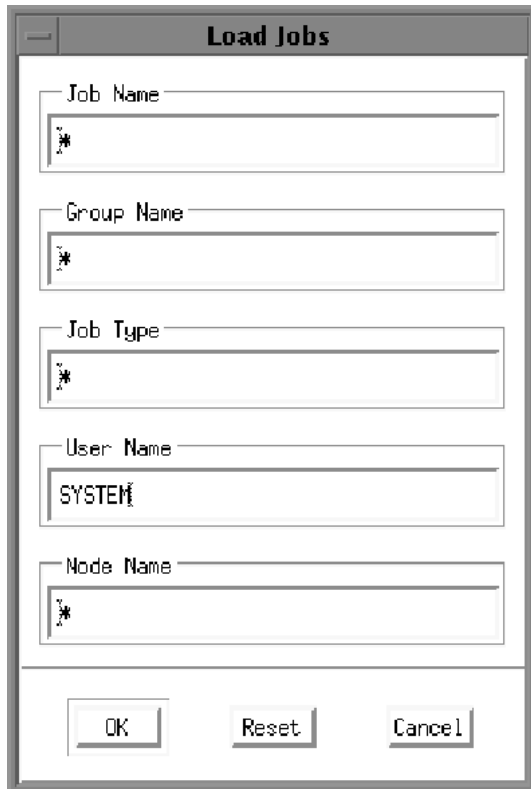
Three windows and a dialog are displayed, as shown in the following figure. The **Graphical Display** window, **Linear Display** window, and **Real Time Messages Display** window have scroll bars for scrolling through displayed information. You can also resize them or minimize them (shrink them to icons).

5.2.4. The Load Jobs Dialog Box

The **Load Jobs** dialog lets you select and display jobs from the **Job Management Manager** database.

You select jobs to display by filling in one or more of the fields in the window. You can use wildcard characters (*) in any of these fields.

The jobs you select from the **Load Jobs** dialog are displayed in both Jobs windows.



The **Load Jobs** dialog box contains five input fields and three buttons. The fields are labeled 'Job Name', 'Group Name', 'Job Type', 'User Name', and 'Node Name'. The 'User Name' field contains the text 'SYSTEM'. Each field has a small icon to its left. At the bottom are three buttons: 'OK', 'Reset', and 'Cancel'.

Clicking **OK** when the fields are filled in as shown in the figure is equivalent to entering either of the following DCL commands:

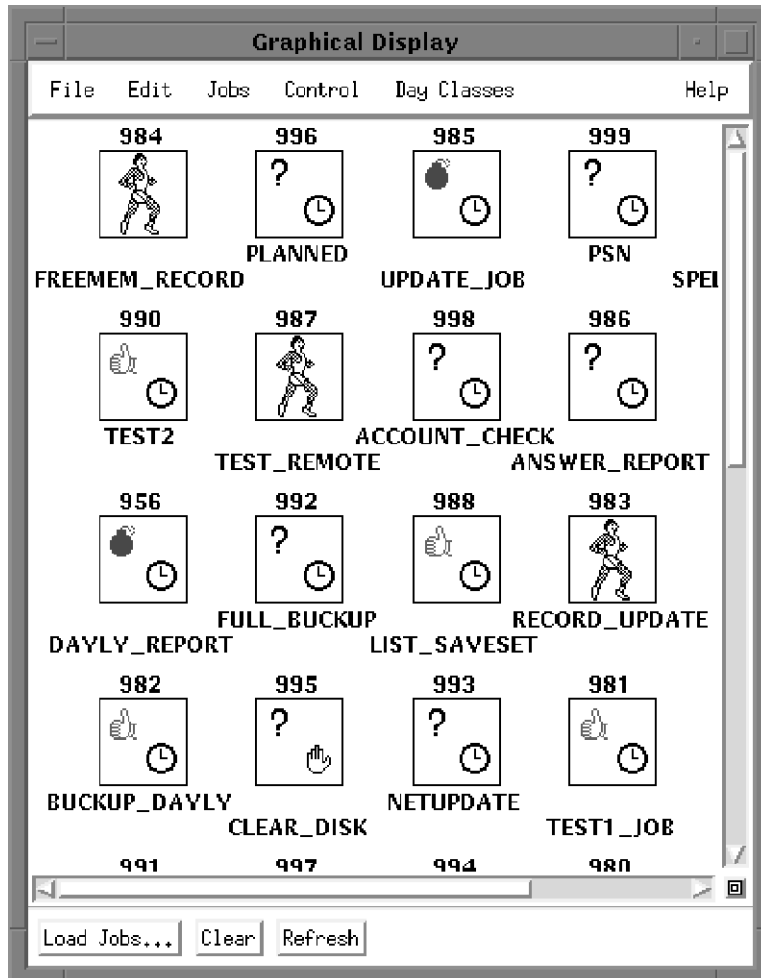
```
$ SCHEDULE SHOW JOBS/USER_NAME=SYSTEM/GROUP=*/TYPE=*/CLUSTER_NODE=*
$ SCHEDULE SHOW JOBS/USER_NAME=SYSTEM
```

The **Load Jobs** dialog is displayed when you start the Motif interface to **Job Management Manager**. You can display this dialog at any time during your session by clicking the **Load Jobs...** button in the Graphical or Linear Display window.

Refer to **Loading Jobs to Display** for more information about how to use the Load Jobs dialog to select jobs to display.

5.2.5. Job Management Manager Graphical Display Window

The **Graphical Display** window provides a graphical display of jobs in the **Job Management Manager** database. Each box shown in this window represents a job. Each job is labeled with its job number preceding and followed by its job name. Refer to **Interpreting Your Job Display** for more information about interpreting your job display.



The **Graphical Display** window also lets you display a **Navigation** window that allows you to select and display portions of the window that are not visible, and to change the size of boxes displayed (see Using Navigation Windows).

The menu bar on this window provides the following five pull-down menus:

- File
 - Print...
 - Quit
- Edit
 - Load Jobs...
 - Clear
 - Load Balance Group Editor...
- Jobs
 - Create...
 - Edit...

- Delete
- Display Output...
- Monitor...
- Control
 - Abort
 - Hold
 - Run
 - Release
- Day Classes
 - Create...
 - Edit...
 - Delete
- Help
 - About

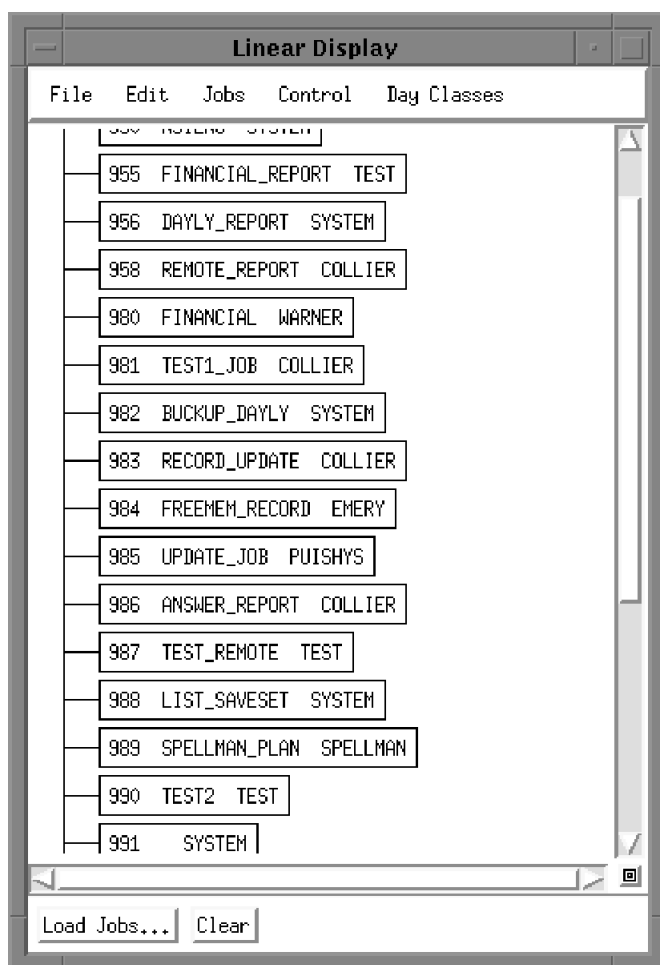
Pressing and holding **MB3** on a job in the **Graphical Display** window displays a pop-up menu with the same options on the Jobs pull-down menu. Moving the pointer over Control and dragging the mouse to the symbol [-> displays a submenu with the same choices found on the Control pull-down menu. To select one of these submenu items, move the pointer over the menu item you want and release **MB3**.

5.2.6. Job Management Manager Linear Display Window

The **Linear Display** window displays a list of the jobs you selected from the **Load Jobs** dialog. Each rectangle in the **Linear Display** window shows the number, name, and owner of a selected job.

By default, all jobs are sorted by job number in the **Linear Jobs Display** window. Refer to **Sorting Jobs** in the **Linear Display** Window for more information about how to sort jobs in the **Linear Display** window.

The **Linear Display** window lets you display a **Navigation** window that allows you to select and display portions of the window that are not visible, and to change the size of boxes displayed (see Using Navigation Windows).



The menu bar in this window includes the same five pull-down menus as the **Graphical Display** window, with the following differences:

- The File menu does not include Print...
- The Edit menu includes the following additional items:
 - Sort by Job number
 - Sort by Job name
 - Sort by Job owner

Pressing and holding **MB3** a job in the **Linear Display** window displays a pop-up menu with the same choices found on the Jobs pull-down menu. Moving the pointer over Control and dragging the mouse to the symbol [-> displays a submenu with the same choices found on the Control pull-down menu. To select one of these submenu items, move the pointer over the menu item you want and release **MB3**.

5.2.7. Job Management Manager Real Time Message Display Window

The **Real Time Messages Display** window displays all **Job Management Manager** job events on the system as they occur during your Job Management Manager session. This window maintains up to 100 lines of messages; you can use the scroll bar to see messages that have scrolled off the top of the window. Messages from previous sessions are stored in a log file with NSCHED

\$.DATA:VERMONT_CREAMERY.LOG as a default name. You need SYSPRV right to view this log file.



Some messages are highlighted, while others are displayed in normal print.

These messages	Provide information about ...
Highlighted	Your jobs that have currently completed, all jobs that were modified, all jobs with control action that were requested
Normal	All other messages

On a color display, different types of messages can appear in different colors, as determined by the file SCHEDULER\$MOTIF.DAT in the DECW\$SYSTEM_DEFAULTS directory or your DECW \$USER_DEFAULTS directory.

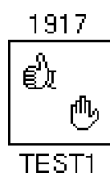
5.3. Interpreting Your Job Display

All job boxes in the **Graphical Display** window contain status symbols. Job boxes may be drawn with a solid or broken line or shaded gray. Some job boxes are connected by arrows, indicating job dependencies. The following sections provide information on interpreting symbols and arrows in the **Graphical Display** window.

5.3.1. Symbols in Job Boxes

Each job box in the **Graphical Display** window contains two symbols. The symbol in the upper-left corner indicates the *last completed status* of that job. The symbol in the bottom-right corner indicates the *current state* of that job.

For example, the hand symbols in the following job box indicate that the job ran successfully last time, and is currently on hold.



The following table describes the symbols displayed in job boxes.

Symbol	Job State	Description
	Failure	The last time the job ran, it failed.
	Success	The last time the job ran, it completed successfully.

Symbol	Job State	Description
	Unknown	The job has never run, so the last completion status is unknown (0).
	Scheduled	The job is scheduled to run.
	Job Slot Wait	The manager is running the maximum number of jobs on all nodes; the job is waiting for execution. One or more other jobs need to finish before this job runs.
	Dependency Wait	The job is waiting for a job dependency to be satisfied before it can run.
	Hold	The job is on hold.
	Running	The job is currently running.
	Queued	The job is scheduled to run in a BATCH queue, and is waiting for execution.

Note

When the job status of an unselected job on a remote node changes, the symbols in the job box are not updated.

5.3.2. Selected and Unselected Job Boxes

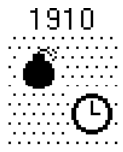
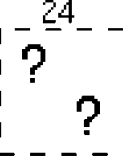
When you select jobs to display in the **Load Jobs** dialog box, the manager also displays any jobs that are connected to the selected jobs by dependencies. These

dependency-related jobs can be on the current node or a remote node. The manager displays the job boxes for selected and unselected jobs differently, as described in the following table. The owner of an unselected job is always displayed.

Note

In this context, unselected jobs are jobs that were not selected for display in the **Load Jobs** dialog box, rather than jobs that were not selected (highlighted) using MB1.

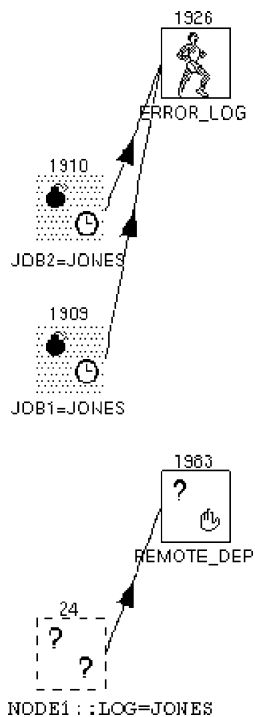
When a job box is displayed like this ...	The job is ...
	Selected It matches the criteria you specified from the Load Jobs dialog box.

When a job box is displayed like this ...	The job is ...
 1910 JOB2=JONES	• Unselected • On the current node • Connected to one or more selected jobs by a dependency
 24 NODE1::LOG=JONES	• Unselected • On a remote node • Connected to one or more selected jobs by a dependency

Note

You cannot perform actions on unselected jobs, nor can you create dependencies that involve them.

The following figure shows an example of selected and unselected job boxes as shown in the **Graphical Display** window:



In the example shown in the figure, a group of jobs was selected. This group contains two jobs:

- ERROR_LOG is dependent on two jobs on the current node.
- REMOTE_DEP is dependent on one job on a remote node.

5.3.3. Job Dependency Arrows

Jobs without dependencies are displayed at the top of the **Graphical Jobs Display** window, and jobs with dependencies are displayed below. Arrows indicate job dependencies. Each arrow indicates the order in which the connected jobs run; the arrow points to the job that depends on the other job.

Note

The manager displays dependent jobs using arrows that point from left to right, in the direction of the flow of execution of the jobs. (Arrows do not always point from left to right in the case of cyclic dependencies.) When you create a dependency that goes from right to left, the manager rearranges the job boxes in the window the next time it displays them, for instance, when you click **Refresh**.

5.4. Loading Jobs to Display

When you start the Motif interface to the manager, load the jobs you want to work with. After you select them and click **OK**, the jobs are displayed in the **Graphical Display** window and the **Linear Display** window.

You can select your own jobs and also jobs that belong to other users. You must have READ access to select a job. You must have WRITE access to modify a job. For more information about READ and WRITE privileges, see Privileges Needed to Use Job Management Manager in the chapter "An Overview of Job Management Manager."

The following figure shows the **Load Jobs** dialog; the table following the figure describes the fields in this dialog.

Field	Description
Job Name	The name of a job. Example: JOB_DAILY
Group Name	The name of a group of jobs. Example: DAILY_BACKUPS

Field	Description
Job Type	The name of a type of job. Example: FRIDAYS
User Name	The name of the owner of the jobs. Example: JONES Your user name is automatically entered in this field when you first access the manager, but you can change the user name to match that of the jobs you want to display.
Node Name	The name of a node within a cluster on which the jobs are restricted to run. Use this field when you want to display jobs that are restricted to running on a specific node.

To select the jobs you want to display:

1. Is the Load Jobs dialog displayed?

- If **Yes**, go to step 2.
- If **No**, click the **Load Jobs...** button in the **Linear or Graphical Jobs Display** window.

The **Load Jobs** dialog is displayed.

2. Click on one or more of the fields in this dialog and enter the information for your job. You can use wildcards (*) in any of the fields.

Characters appear in the field as you type.

3. When you are finished entering the information, click **OK** or press Return.

The **Load Jobs** dialog is removed, and the jobs that match the criteria you specified are displayed in **Display** windows.

5.5. Clearing and Changing the Currently Displayed Jobs

If you want to clear the displayed jobs and display new jobs, you must first clear the **Display** windows and then select new jobs to display, as follows:

1. Click the **Clear** button in the **Graphical Display** window or **Linear Display** window.

Note

If you do not click the **Clear** button, any new jobs that you select are added to the jobs currently displayed.

All jobs are removed from both **Display** windows.

2. Click the **Load Jobs...** button in either window. The **Load Jobs** dialog box is displayed.
3. Click on one or more fields in the dialog box and edit information for new jobs. You can move the text cursor and insert characters, use the delete key, or you can highlight characters and use pending delete. You can clear all fields in the dialog box by clicking **Reset**.

The characters appear in the fields as you type.

4. Click **OK**.

The new jobs that match the criteria you specified are displayed in both **Display** windows.

5.6. Adding to the Currently Displayed Job

To add more jobs to those currently displayed, follow these steps:

1. Click the **Load Jobs...** button in either **Jobs Display** window. The **Load Jobs** dialog box is displayed.
2. Click on one or more fields in the dialog box and edit information for new jobs. You can move the text cursor and insert characters, use the delete key, or you can highlight characters and use pending delete. You can clear all fields in the dialog box by clicking **Reset**. See the table in **Loading Jobs to Display** for a description of the fields.

The characters appear in the fields as you type.

3. Click **OK**.

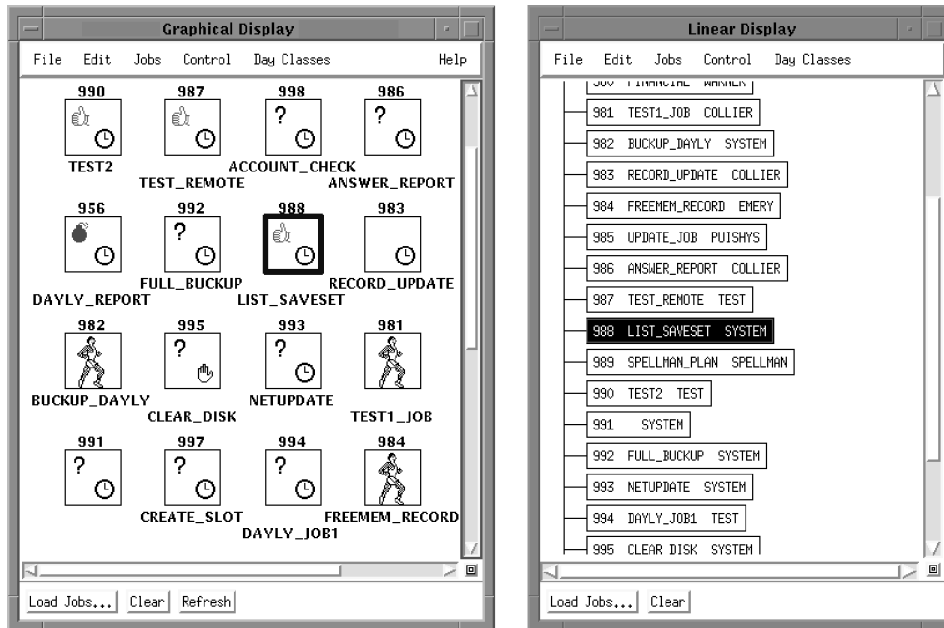
The new jobs that match the criteria you specified are added to both **Display** windows; you may need to use scroll bars to locate the new jobs.

5.7. Locating Jobs by Scrolling

If you have selected many jobs in your VSI Management Manager database, you cannot be able to view all of your jobs at one time in the **Display** window. To perform an action on a job, first locate it in the **Display** window and then select it by clicking it with **MB1**.

Scroll bars to find the job in the **Graphical Display** window and click it with **MB1**. In the **Linear Display** window, you can sort jobs by number, job name, or job owner using the **Edit** menu.

When you select a job by clicking it, it is highlighted in both **Display** windows, as shown in the following figure. When you click a job in the **Linear Display** window, the **Graphical Display** window scrolls automatically to show the job, if necessary. The following figure shows both **Display** windows, with the job LIST_SAVESET selected:



5.8. Viewing Job Information

The Motif interface lets you view three types of information about a job:

Information Type	Jobs Menu Item	Description
Attributes	Edit...	Displays the attributes of a job.
Output file	Display Output...	Displays the output of a job.
Resource usage	Monitor...	Displays and updates the resource usage of a job while the job is running.

5.8.1. Viewing a Job's Attributes

To view the attributes of a job, select it using **MB1** and choose **Edit...** from the **Jobs** menu, or choose **Edit...** from the job's pop-up menu. In the **Graphical Jobs Display** window, you can also Double-click on the job using **MB1**. The manager displays the job in the **Job Edit** dialog box. You can also edit the job from this dialog box (see Editing a Job).

More information about a job is available by clicking on the buttons at the bottom of the **Job Edit** dialog box: **Auxiliary Options...**, **View Deps...**, and **Day Class Restrictions...**

Viewing the attributes of a job is similar to using the following command in the DCL interface:

```
$ SCHEDULE SHOW JOB job_name/FULL
```

Note

To avoid changing a job's attributes, be sure to click the **Cancel** button in the **Job Edit** dialog box and its associated additional dialog boxes; do not click **OK** or **Apply**.

The following figure shows the primary **Job Edit** dialog box:

5.8.2. Viewing Job Output Files

A job output file captures the output of a job. An output file is not created unless a file name has been entered in the **Output File** field of the job's **Job Edit** dialog box. When you view the output file for a job, the manager creates a window to display the output log file. You can shrink this window to an icon.

To view the output file for a job, select the job and choose **Display Output...** from the **Jobs** menu. When you are finished viewing the information, click the **Dismiss** button to remove the window.

5.8.3. Viewing Job Resource Usage

The **Job Monitor** window lets you monitor a job's resource usage based on its historical minimum, average, and maximum usage. To display the **Job Monitor** window, select a job and choose **Monitor...** from the **Jobs** menu. You can shrink this window to an icon. The following figure shows the **Job Monitor** window:

Job	3 - FINANCIAL_REPORT					
User	SYSTEM					
Count	4 Success Records only					Cancel
From	15-JAN-1992 13:06:41					
Thru	11-FEB-1992 18:02:00					
Status	Job not running					
		Cur	Min	Ave	Max	Scale
Elapsed		0 00:00:00	13.0	13.0	13.0	Sec
CPU		0 00:00:00	1.0	1.0	1.0	Sec
Pg Flts		0	560.0	566.0	569.0	
Buff I/O		0	43.0	43.0	43.0	
Dir I/O		0	92.0	95.0	97.0	

The top half of the **Job Monitor** window lists job information such as the job name and number, user name, success count, and status. On the bottom-left are histograms that graphically display the resource

usage of the job. For each job running in the **Display** window, you can monitor the current resource usage of the following:

- Elapsed time
- CPU time
- Page faults
- Buffered I/O count
- Direct I/O count

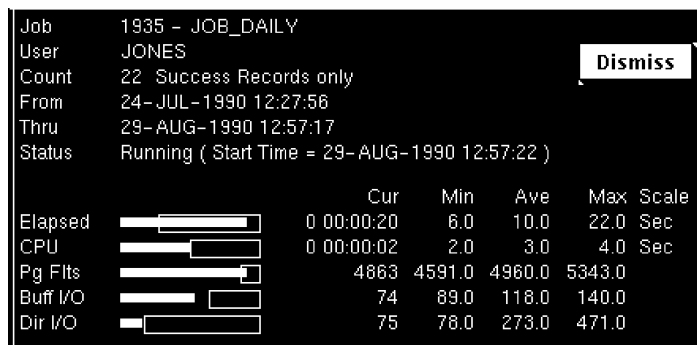
On the bottom-right of the window are the actual numbers associated with the histograms.

The rectangle outlines next to each resource indicate the historical usage. The left side of each rectangle represents the minimum usage, the right side represents the maximum usage. When information is within a narrow range, a thin or thick line is displayed instead of a rectangle.

The right sides of the rectangles are positioned relative to the maximum usage of each resource. Therefore, when a job is not running, all rectangles will be even on the right side.

5.8.3.1. While the Job Is Running

While the job is running, white or colored bars display next to each resource to indicate the current usage; the rectangle outlines remain present to indicate the historical usage. While the job is running, the **Job Monitor** window automatically refreshes itself every 10 seconds to update the resource usage information.



5.8.3.2. When a Job Exceeds Historical Usage

When a job reaches the historical maximum usage for a resource, the white bar and the rectangle outline next to that resource are even on the right side. As the job begins to exceed the maximum usage for a resource, the rectangle outline next to that resource shifts to the left to show the current maximum usage.

The following figure shows a job that has exceeded the historical maximum usage for all resources:

Job	1921 - DAILY_JOB					
User	JONES					
Count	10 Success Records only					
From	31-AUG-1990 11:59:04					
Thru	31-AUG-1990 12:41:13					
Status	Running (Start Time = 31-AUG-1990 12:44:06)					
		Cur	Min	Ave	Max	Scale
Elapsed		0 00:00:51	8.0	24.0	45.0	Sec
CPU		0 00:00:25	3.0	11.0	22.0	Sec
Pg Flts		10878	5401.0	7397.0	10244.0	
Buff I/O		771	128.0	345.0	671.0	
Dir I/O		14133	389.0	4692.0	11723.0	

Note how far the rectangles have shifted to the left. This distance allows you to compare the current usage (indicated by the white or colored bar), with the historical maximum usage (indicated by the rectangle outline).

5.9. Creating a Job

Use the **Job Create** dialog box to create a job. This dialog box contains text entry fields for entering job attributes and buttons for displaying three additional related dialog boxes. The three buttons at the bottom of the dialog box are grayed out until you create the job by clicking the **Apply** button.

CA Job Management Job Create (Primary)

Job Name:

User Name:

Group:

Type:

Comment:

Mail Address:

Job State:

Next Run Time:

Schedule Interval:

Last Start:

Last Completion:

Successful Runs: 0 Failed: 0

Command: ...

Pre-Function: ...

Post-Function: ...

Output File: ...

Max Run Time: Max Time Job #:

Max Stall Time: Stall Time Job #:

Job Security

Rights Identifiers

Read:

Write:

Execute:

Additional options

Auxiliary Options...

View Dependencies...

Day Class Restrictions...

OK Apply Cancel

The following figure shows the primary **Job Create** dialog box, which is similar to the **Job Edit** dialog box:

To create a job in the **Job Management Manager** database:

1. Pull down the **Jobs** menu from the **Graphical** or **Linear Display** window and choose **Create...**

The **Job Create** dialog box is displayed.

- Click on fields described in the following table and enter the information necessary for your job. Note that you can enter information only in the fields described in the following table. The manager automatically fills in the remaining fields after the job runs for the first time.

Characters appear in the fields as you type.

- When you have finished entering information, click **OK** to add your new job to the **Job Management Manager** database.

A highlighted message reporting the job creation is displayed in the **Real Time Message Display** dialog box, and a small DECwindows message box also appears.

- Click **OK** in the small DECwindows message box. The **Job Create** dialog box is removed from the screen.

The following table describes the fields you can use to enter job attributes in the **Job Create** dialog box.

Use this field...	To enter...
Job Name	The name of your job.
User Name	Your OpenVMS user name.
Group	The name of the group that this job belongs to.
Type	The name of the job type that this job belongs to.
Comment	Any miscellaneous comments you want to add.
Command	The primary system-level command to perform when this job is run.
PreFunction	A command to perform before the job's primary command.
PostFunction	A command to perform after the job's primary command.
Output File	The name of the file to receive the output of this job.
Schedule Interval	The Job Management Manager delta time interval for running the job. Example: D 01:00 specifies to run the job daily at 1:00 A.M.
Stall Time Job #	A local job to execute if the job's stall time is exceeded. Note that you must use Job Management Manager delta time in this field (+DDDD HH:MM:SS.)
Max Time Job #	A local job to execute if the job's maximum time is exceeded. Note that you must use Job Management Manager delta time in this field (+DDDD HH:mm:ss).
Read Identifier	A Privilege identifier that overrides the normal UIC protection.
Write Identifier	A Privilege identifier that overrides the normal UIC protection.
Execute Identifier	A Privilege identifier that overrides the normal UIC protection.
Next Run Time	When to run the new job for the first time Default: Never.
Maximum Run Time	The maximum Job Management Manager delta time for the job to run before the Max Time Job is started.

Use this field...	To enter...
	Default: None. Use the format +DDDD HH:MM:SS. You cannot use this field if you do not specify a Max Time Job.
Stall Time	The maximum Job Management Manager delta time for the job to stall before the Stall Time Job is started. Default: None. Use the format +DDDD HH:MM:SS. You cannot use this field if you do not specify a Stall Time Job.
Mail Address	The address to which to send Mail. Default: your username.

After you have filled in the fields' necessary to create your job, you can click **Apply** to create the job and then click **Auxiliary Options...** and/or **Day Class Restrictions...**, if you need to use these dialog boxes to enter more information.

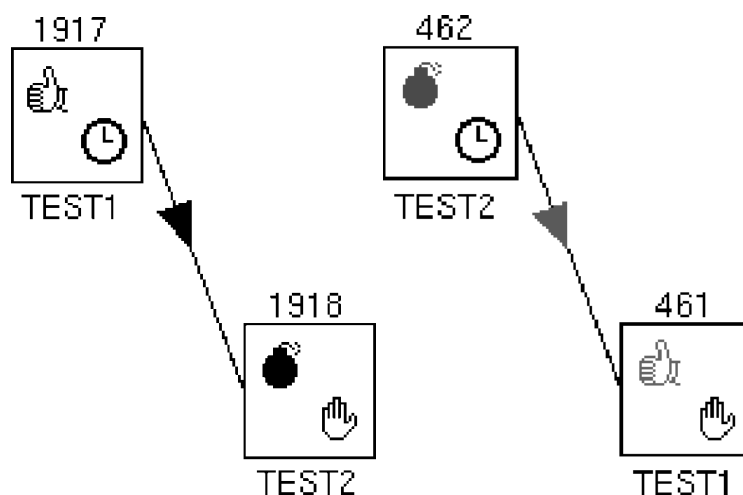
5.10. Creating a Job Dependency

When you create a dependency, the direction in which you drag the mouse determines the dependency between two jobs: you press **MB1** on the job that must complete before the job that depends upon it, drag, and release **MB1** on the box that contains the dependent job. In example 1, the mouse was dragged from TEST1 to TEST2. In example 2, the mouse was dragged from TEST2 to TEST1. The arrow points in the direction of the dependent job.

Note

The manager displays dependent jobs using arrows that point from left to right, in the direction of the flow of execution of the jobs. When you create a dependency that goes from right to left, the manager rearranges the job boxes in the window the next time it displays them, for instance, when you click **Refresh**.

Example 1 Example 2



To create a dependency for a job:

1. Locate the job you want to run first in the **Linear Display** window.
2. Select the job by clicking **MB1** on it.

The job is highlighted in the **Graphical Display** window and the **Linear Display** window.

3. In the **Graphical Display** window, move the pointer over the job that you want to run first. Press and hold **MB1**.
4. Is the second job visible in the **Graphical Display** window? If Yes, drag the mouse to the second job and release **MB1**.

As you drag the mouse, you create a line that follows the pointer from the first job to the second job. When you release **MB1**, an arrowhead appears on the line, pointing from the first job to the second job. Messages appear in the **Real Time Messages Display** window reporting that the second job was modified, and by whom.

5.11. Deleting a Job Dependency

To delete a job dependency, press and hold **MB3** over the arrow that indicates the dependency and choose **Remove Dependency** from the pop-up menu. Messages appear in the **Real Time Message Display** window reporting that a dependency was deleted, and by whom.

To cancel this operation while in progress, move the pointer outside the pop-up menu and release **MB3**.

5.12. Controlling a Job: Aborting, Holding, Running, and Releasing

To abort, hold, run, or release a job:

1. Select the job by clicking **MB1** on it.

The job is highlighted in both **Display** windows.

2. Pull down the **Control** menu in either **Jobs Display** window and choose the action you want.

The *current state* symbol in the lower-right corner of the job box changes to reflect the current state of that job.

For example, if you choose **Hold**, the *current state* symbol changes to an open hand. See the table in Symbols in **Job Boxes** for a list of the job status symbols.

Messages appear in the **Real Time Messages Display** window reporting the action you have taken.

5.13. Editing a Job

Use the **Job Edit** dialog to edit jobs; this dialog contains text entry fields for entering job information, and buttons for displaying three additional related dialog boxes. This dialog is similar to the **Create Job** dialog.

To edit a job, select the job and click **Edit...** from the **Jobs** menu. In the **Graphical Display** window, you can also Double-click on a job to edit it. The following figure shows the primary **Job Edit** dialog:

The screenshot shows the 'CA Job Management Job 2 Edit (Primary)' dialog box. It has a title bar with standard window controls. The main area is divided into several sections:

- Job Attributes:** Fields for Job Name (TEST), User Name (SYSTEM), Group (TEST), Type, Comment (Test), and Mail Address (@MYMAILLIST.DIS).
- Job State:** Scheduled.
- Next Run Time:** Never.
- Schedule Interval:** (empty field).
- Last Start:** 29-JUL-2008 19:18:33.56.
- Last Completion:** 29-JUL-2008 19:18:36.87.
- Successful Runs:** 8.
- Failed:** 0.
- Command:** spawn mail/noself/subj:"my subject" NL: "@mymailist.dis" (with a browse button).
- Pre-Function:** @TCPIP\$DEFINE_COMMANDS (with a browse button).
- Post-Function:** (empty field with a browse button).
- Last Exit Status:** Success (with a dropdown menu).
- Output File:** (empty field with a browse button).
- Max Run Time:** None (with a dropdown menu).
- Max Time Job #:** (empty field).
- Max Stall Time:** None (with a dropdown menu).
- Stall Time Job #:** (empty field).
- Job Security:** Fields for Read, Write, and Execute permissions.
- Additional options:** Buttons for Auxiliary Options..., View Dependencies..., and Day Class Restrictions...
- Buttons:** OK, Apply, and Cancel (the Cancel button is highlighted with a red rectangle).

This dialog contains text entry fields for entering job attributes and buttons for displaying three additional related dialog boxes.

To edit a job in the **Job Management Manager** database:

1. Pull-down the **Jobs** menu from the **Graphical** or **Linear Display** window and click **Edit...**

The primary **Job Edit** dialog is displayed.

2. Click the field you want to edit and enter new information for the job (repeat as required). To edit **Auxiliary Options** or **Special Day Restrictions**, click the corresponding button to bring up an additional dialog.

Characters appear in the fields as you type.

3. When you have finished editing information in a dialog, click **OK**. If you want to save changes in the primary dialog and go on to other associated dialog boxes, click **Apply**.

A highlighted message reporting the job modification is displayed in the **Real Time Message Display** dialog, and a small DECwindows message appears.

4. Click **OK** in the small DECwindows message.

The associated dialog is removed from the screen.

Refer to the table in Creating a Job for a description of fields in the primary **Job Create** dialog.

Unicenter Job Management Job 2 Edit (Auxiliary)

Execution Mode

- ☒ Detached
- ☐ OpenVMS Batch
- ☐ Remote

Valid Run Days

- ☐ Sunday
- ☐ Monday
- ☐ Tuesday
- ☐ Wednesday
- ☐ Thursday
- ☐ Friday
- ☐ Saturday

Retain Job

- ☒ Always
- ☐ Never
- ☐ On Success
- ☐ On Failure

Notify by Mail

- ☒ Never
- ☐ Always
- ☐ On Success
- ☐ On Failure

OPCOM Messages

- ☒ No
- ☐ Yes

Detached Process Attributes

Cluster Node Restriction:

Run Priority: (0 to use Job Management DEFAULT)

OpenVMS Batch Job Attributes

Batch Queue:

Batch Queue Entry #:

Time Submitted:

CPU Limit (X 100ms):

Batch Queue Priority:

Remote Job Attributes

Remote Execution Information (USER@NODE):

Load Balance Group

G2

Notify Terminal

- ☒ Always
- ☐ Never

Retry Job

- ☒ Never
- ☐ On Failure

Add Delta Time Intervals

- ☒ To Actual Job Start Time
- ☐ To Scheduled Job Start Time

Notify by Mail on Special Day Action

- ☒ No
- ☐ Yes

Restart Job

- ☒ On Crash
- ☐ Never

The following figure shows the **Auxiliary Options** dialog:

For information about **Special Day Restrictions**, see the section Setting or Editing Day Class Restrictions.

For information about Load Balance Groups, see the section Setting or Editing Load Balance Groups for Jobs.

5.14. Deleting a Job

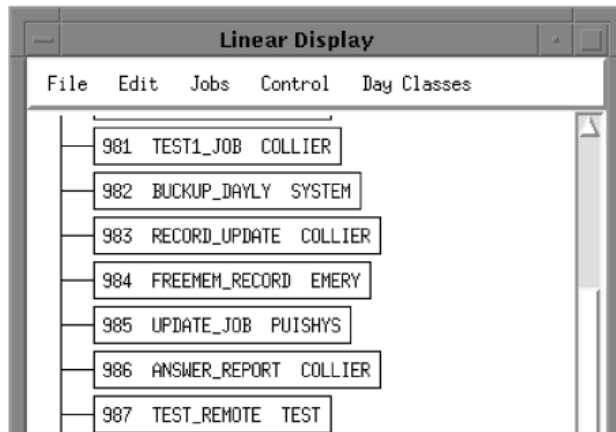
To delete a job from the Job Management Manager database, select the job using **MB1** and click **Delete** from the **Jobs** menu. After you delete a job, click **Refresh** to reposition job boxes in the **Graphical Display** window.

5.15. Sorting Jobs in the Linear Display Window

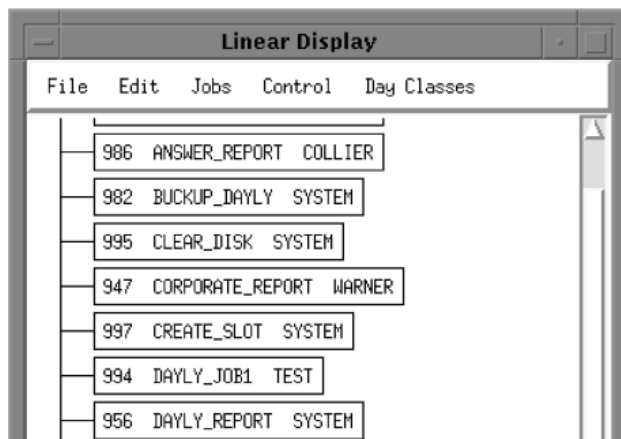
By default, the manager sorts jobs by job number in the **Linear Display** window. You can also sort by job name or job owner, as well as resort by job number. Note that only the jobs in the Jobs window are sorted.

To sort the jobs in the **Linear Display** window, choose the type of sorting you want from the **Edit** menu.

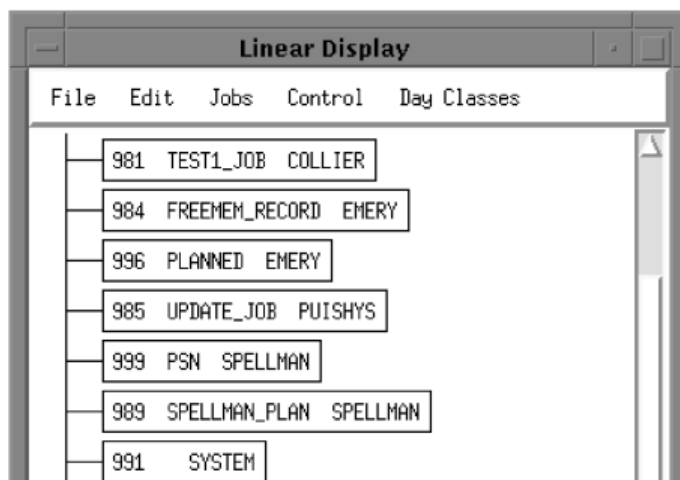
Example of Jobs Sorted by Job Number



Example of Jobs Sorted by Job Name

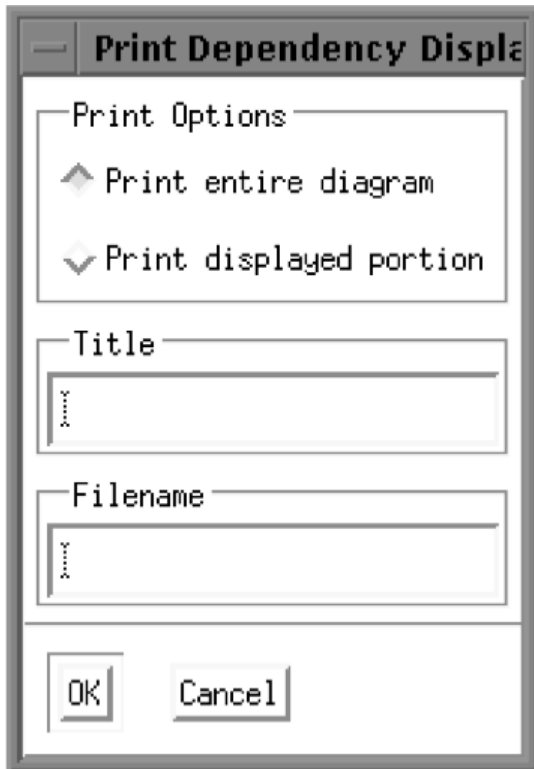


Examples of Jobs Sorted by Job Owner



5.16. Creating a Graphics File of Your Display

You can save all or part of the current **Graphical Display** window as a graphics file. When you choose **Print...** from the **File** menu in the **Graphical Display** window, the **Print Dependency Display** dialog box appears.



Click on one of the two option buttons to choose the entire diagram or the displayed portion only. Enter a title and a filename, and then click **OK**. (The title will appear at the top of the printed file.)

The manager creates a **.DDIF** file and places this file in your top-level directory if you do not specify a directory. You can convert the **.DDIF** file into another format for printing, such as PostScript or sixel graphics, using the DCL command **CONVERT**:

```
$ CONVERT/DOCUMENT filename.DDIF/FORMAT=DDIF
_To: filename.type/FORMAT=type
```

Note

Job status symbols do not appear in the job boxes in the **.DDIF** file.

5.16.1. Using Navigation Windows

A **Navigation** window is a small window that lets you view the overall contents of the larger window, including the areas not currently displayed. You can see the number of selected jobs not currently visible in the larger windows. **Navigation** windows do not display the text displayed in the larger windows. You cannot shrink **Navigation** windows to icons.

You can display a **Navigation** window for either the **Graphical** or **Linear Display** window by clicking on the box located in the corresponding window where the horizontal and vertical scroll bars meet.

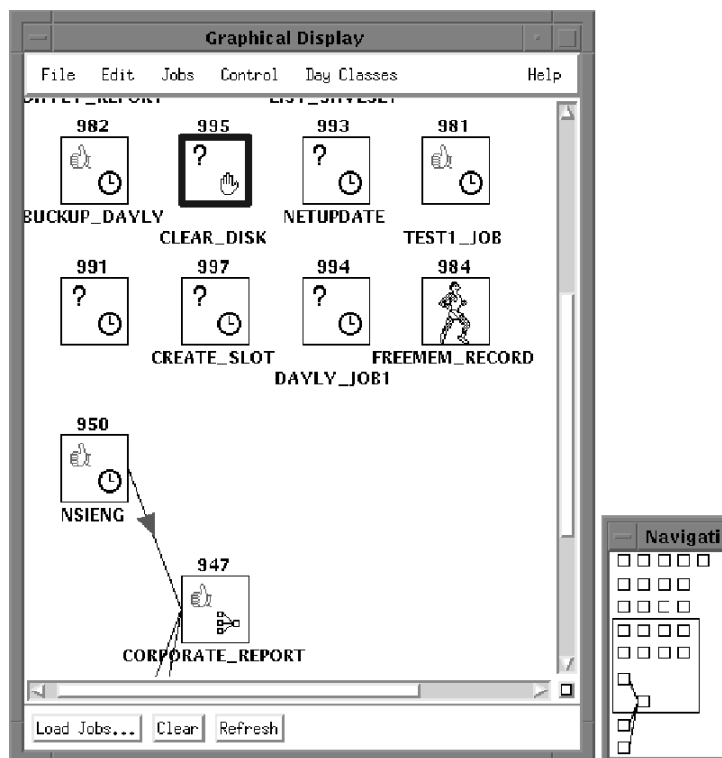
When you have finished using the **Navigation** window, click on this box again to remove the **Navigation** window from the screen.

The following sections describe how to use a **Navigation** window to:

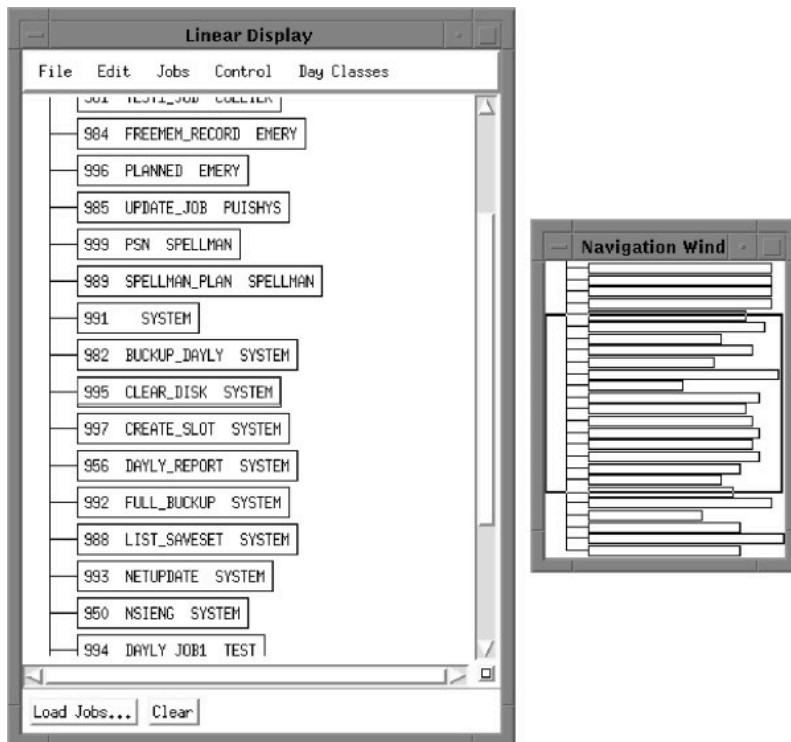
- Select and display a new portion of the **Graphical Display** window or the **Linear Display** window
- Scale objects in the **Graphical Display** window

The next two figures show the **Navigation** windows for the **Graphical Display** window and the **Linear Display** window.

5.16.1.1. Graphical Display Window and Its Navigation Window



5.16.1.2. Linear Display Window and It Navigation Window



5.16.2. Selecting and Displaying a New Portion of a Display Window

In a **Navigation** window, the contents of the main window are displayed within a rectangular outline in the **Navigation** window. To select and display a different area of the main window, click inside this outline and use **MB1** to drag it to the new location inside the **Navigation** window. The main window scrolls to show the new area.

5.16.3. Resizing a Portion of the Grapical Display Window

To scale objects in the **Graphical Display** window (for instance, to see more job boxes in a smaller area):

1. Move the pointer within the rectangle outlined in the **Navigation** window. Then press and hold **MB3**.

The pointer changes to a + symbol when it is placed in the rectangle.

2. Drag **MB3** slowly toward the edge of the rectangle to display a larger area with smaller job boxes, or drag **MB3** toward the center of the rectangle to display a smaller area with larger job boxes.

As you drag **MB3**, the size of the rectangle changes.

3. Release **MB3**.

The job boxes in the **Graphical Display** window are resized.

Note the **Graphical Display** window does not change size, but the job boxes are resized. Also note job status icons are not scaled.

- Repeat steps 2 through 4 until the desired size is achieved.

5.17. Special Day Classes

In the Motif interface, use the **Day Classes** menu in either **Display** window to perform the following **Special Day** functions:

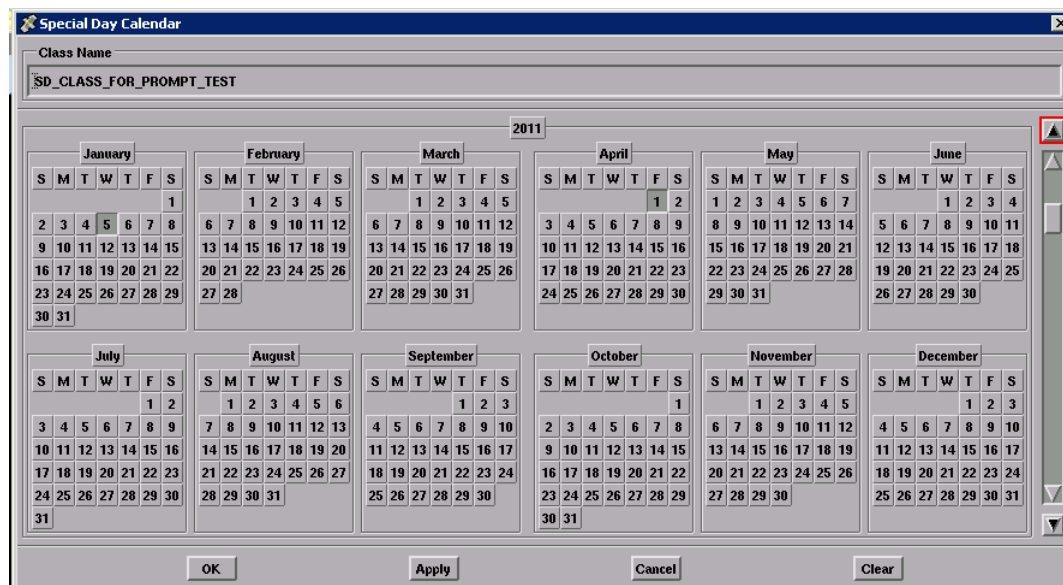
- Create Special Day Classes
- Edit Special Day Classes
- Delete Special Day Classes

For general information about Special Day, see Special Days in the chapter "How Job Management Manager Works."

Use the **Day Class Restrictions** dialog to set Day Class Restrictions for a job; you can access this dialog from the primary **Job Create** or primary **Job Edit** dialog by clicking the **Day Class Restrictions** button.

5.17.1. Creating a Special Day Class

Creating Special Day Classes requires either OPER or SYSPRV privileges.



You can create a Special Day Class by choosing **Create...** from the **Day Classes** menu. This action displays the **Special Days Calendar** dialog; you select the days that you want to make up the class by clicking them in the calendar. The shaded days on the calendar define the **Special Day Class**.

You can switch the entire year, complete months, or individual days by clicking **MB1** on them. You can switch all monthly occurrences of a day by clicking its symbol. For example, to select all Wednesdays in May, click the **W** under **May** on the **Calendar** screen.

Select any year from 2010 through 2019 by clicking up or down on the scroll bar to the right of the calendar. You can scroll to the next decade by dragging the scroll bar to the bottom (2019) and then clicking **MB1** on the stepping arrow beneath the scroll bar. You can scroll to the previous decade by dragging the scroll bar to the top (2010) and then clicking **MB1** on the stepping arrow above the scroll bar. You can use the scroll bar and arrows to navigate through the years 1970-2037.

To create Special Day Classes with the Motif interface:

1. Click **Create** from the **Day Classes** menu.

The **Job Management Manager Calendar** is displayed.

2. Click the day or days that comprise the class. At any point, you can click **Cancel** or **Clear**.

The selected days are shaded.

3. When you have finished defining the class, enter its name in the text field labeled **Save as class name**.
4. Click **OK** to save the class data. The Special Days Class is created.

5.17.2. Editing a Special Day Class

To edit a Special Day Class with the Motif interface:

1. Choose **Edit...** from the **Day Classes** menu.

A list box of existing Day Classes is displayed as shown in the following figure:



2. Double-click on the name of the class to edit. The Special Days Calendar is displayed.
3. Edit the class by toggling days, months, or years.

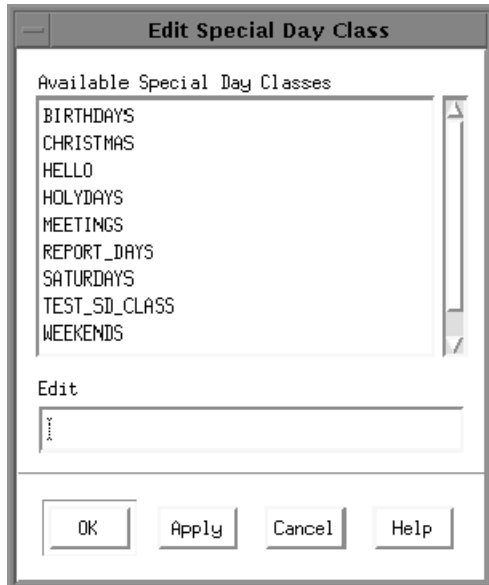
The new Special Day Class is defined by the days that are shaded.

4. Click **OK** to save changes to the class data. The Special Days Class is redefined.
5. A dialog box appears that asks you to confirm the changes. Click **OK** to save the changes.

5.17.3. Deleting a Special Day Class

To delete a Special Day Class with the Motif interface:

1. Choose **Delete...** from the **Day Classes** menu.



A list box of existing **Day Classes** is displayed as shown in the following figure:

2. Double-click the **Special Day Class** to delete.

A dialog box appears that asks you to confirm the deletion.

3. Click **Yes** to confirm the deletion, or **No** to abort the deletion. The dialog box is cleared from the screen.

5.17.4. Setting or Editing Day Class Restrictions

Day Class Restrictions let you restrict a job so that it:

- Runs only on the days defined by one or more Special Days Classes
- Does not run on the days defined by one or more Special Days Classes

To enter or edit Day Class Restrictions for a job, follow these steps:

1. Double-click on the job in the **Graphical Display** window, or select the job in either **Display** window and choose **Edit...** from the **Jobs** menu.

The primary **Job Edit** dialog box appears.

2. Click the button labeled **Day Class Restrictions...** in the dialog box. A dialog box appears for entering Day Class Restrictions.
3. Double-click on available **Special Day Classes** in the left-hand list box to transfer them to the right-hand list box of classes by which the job will be restricted.

The class names move to the right-hand list box.

4. You can also remove Special Day Classes individually from the right-hand list box by Double-clicking on them there, or clear all restrictions by clicking **Clear Restrictions**.

The class names move to the left-hand list box.

5. After you have chosen all Special Day Classes you want to apply to the job, click on one of the two left-hand radio buttons to choose whether the job will not run on included days, or will run only on included days.

The radio button is highlighted.

6. Click on one of the two right-hand radio buttons to choose whether the job will be skipped on a Special Day or placed on hold until operator intervention.

The button is highlighted.

7. Click **OK** to set the Day Class Restrictions.

A DECwindows message box appears informing you that the job has been modified.

8. Click **OK** in the DECwindows message box.

The Day Class Restrictions dialog box is cleared from the screen.

9. If you want to notify a job owner by Mail when a Special Day Action occurs, click **Auxiliary Options...** in the primary **Job Edit** dialog box, and then click on the appropriate radio button at the bottom of the **Auxiliary Options** dialog box.

The button is highlighted.

10. Click **OK** in the **Auxiliary Options** dialog box. A DECwindows message box appears.

11. Click **OK** in the DECwindows message box.

The **Auxiliary Options** dialog box is cleared from the screen.

12. Click **OK** in the primary **Job Edit** dialog box. The dialog box is cleared from the screen.

Note

- You cannot use the Apply button repeatedly to combine different restrictions, such as running for some Special Days Classes and not running for others.
- Setting or editing a job's Day Class Restrictions does not affect its next scheduled run time. The restrictions apply to run times following the next scheduled run time.

5.18. Load Balance Groups

In the Motif interface, the **Load Balance Group Editor...** option in the **Edit** menu of either display window lets you perform the following Load Balance Group functions:

- Create Load Balance Groups
- Edit Load Balance Groups

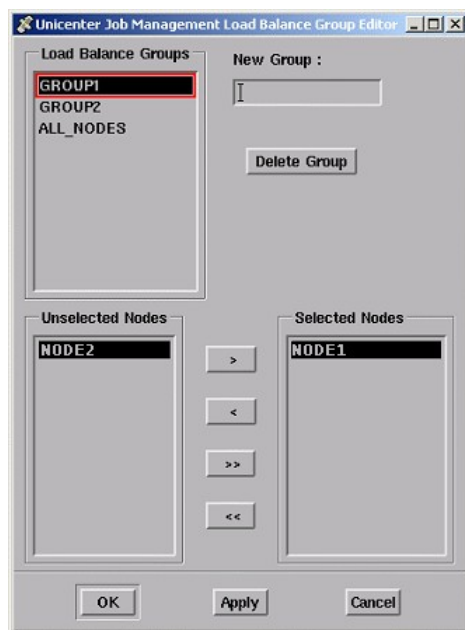
- Delete Load Balance Groups

For more information on Load Balance Groups, see the chapter "How Job Management Manager Works."

The **Auxiliary Job Edit** dialog box lets you assign a Load Balance Group to a job or remove an assignment. You access this dialog box from the primary **Job Create** or primary **Job Edit** dialog box by clicking the **Auxiliary Options...** button. You can also enter the Load Balance Group Editor while editing a job. From the **Auxiliary Job Edit** dialog box, click the **Groups Editor** button.

5.18.1. The Load Balance Group Editor

You can create, edit, or delete a Load Balance Group in the Load Balance Group Editor. To access the Load Balance Group Editor, select **Load Balance Groups Editor...** from the **Linear Display** or **Graphical Display Edit** menu. The **Load Balance Group Editor** appears.



From the editor, you can select the nodes from your local cluster that you want to include as members of the Load Balance group. The dialog box includes a list of Load Balance Groups. For the currently selected Load Balance group, the editor displays a list of **Unselected Nodes** and a list of **Selected Nodes**. With the editor, you can also delete a group and specify a new group name.

5.18.2. Creating a Load Balance Group

To create Load Balance Groups, you must have either OPER or SYSPRV privileges.

To view the nodes that are members of a Load Balance group, click the group name in the Load Balance Groups list. You can move a node from the **Unselected Nodes** list to the **Selected Nodes** list by clicking on the node name and then clicking on the ">" button. The ">>" button lets you move all **Unselected Nodes** to the **Selected Nodes** list. The "<" and "<<" buttons remove nodes from the **Selected Nodes** list.

To create Load Balance Groups with the Motif Interface, follow these steps:

1. Select **Load Balance Groups Editor...** from the **Edit** menu. You can also click the **Groups Editor** button from the **Auxiliary Job Edit** dialog box.

2. Enter the group name in the **New Group** field. Click **Apply**.
3. Select nodes from the **Unselected Nodes** list.

The selected nodes appear in the **Selected Nodes** list.

4. When you have finished defining the Load Balance Group, click **OK**. The Load Balance Group is created.

5.18.3. Editing a Load Balance Group

To edit a Load Balance Group with the Motif interface, follow these steps:

1. Select **Load Balance Groups Editor...** from the **Edit** menu. You can also click the **Groups Editor** button from the **Auxiliary Job Edit** dialog box.
2. Click the Load Balance Group name that you want to edit.
3. As needed, move nodes between the **Unselected Nodes** list and the Selected nodes list.

The selected nodes appear in the **Selected Nodes** list.

4. When you have finished editing the Load Balance Group, click **OK**. The changes to the Load Balance Group are saved.

5.18.4. Deleting a Load Balance Group

To delete a Load Balance Group with the Motif interface, follow these steps:

1. Select **Load Balance Groups Editor...** from the **Edit** menu. You can also click the **Groups Editor** button from the **Auxiliary Job Edit** dialog box.
2. Click the Load Balance Group name that you want to delete.
3. Click **Delete Group**. The Load Balance Group is deleted.

Note

If the group is referenced in any job definitions, the reference is removed, leaving the job with no Load Balance Group assignment. For more information, see *Chapter 3, "Getting Started with Job Management Manager"*.

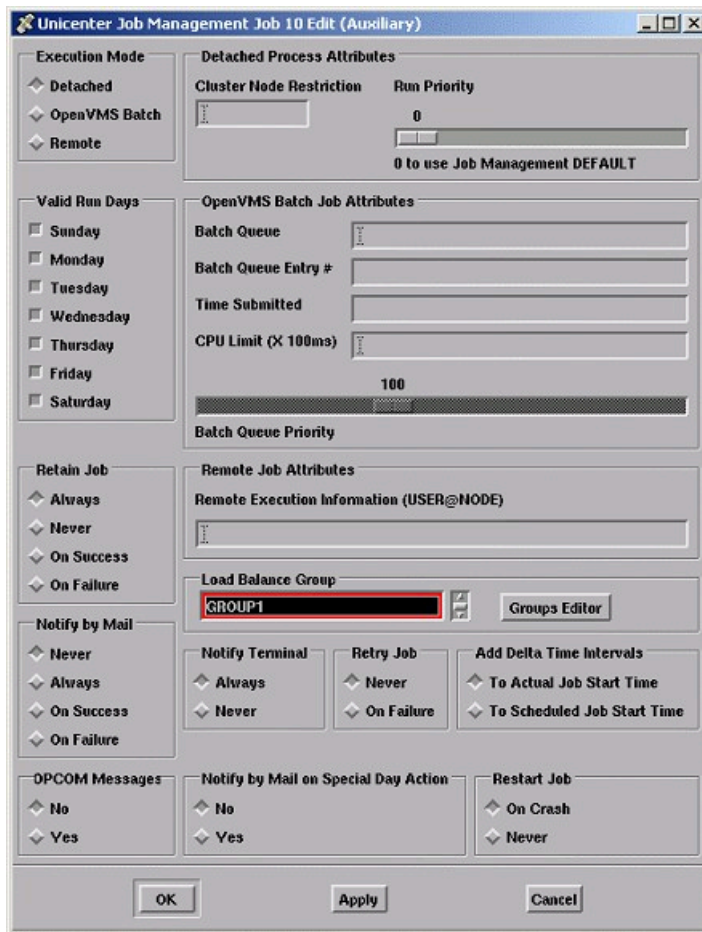
4. Click **OK**. The changes to the Load Balance Group are saved.

5.18.5. Setting or Editing a Load Balance Group for a Job

To assign a Load Balance Group to a job or change the Load Balance Group assigned to a job with the Motif Interface, follow these steps:

1. Double click on the job's icon in the **Graphical** or **Linear Display**. You can also select the job by clicking the job and then select **Edit...** from the jobs menu.

- Click the **Auxiliary Options...** button. The **Job Management Manager Job 10 Edit (Auxiliary)** dialog appears.



- Use the arrows to locate the Load Balance Group.

Note

If you need to create a new group, you can click the **Groups Editor** button.

- Click the name of the Load Balance Group you want to assign the job to. To remove a Load Balance Group assignment, select **None**.
- Click **Apply**. Your selection is saved. You can also click **OK** to save your selection and exit the **Job Management Manager Job 10 Edit (Auxiliary)** dialog.

5.19. Exiting from the Motif Interface

To exit from the Motif interface, pull down the **File** menu from the **Graphical Display** window or **Linear Display** window and select **Quit**. All Job Management Manager windows are removed from the screen.

Appendix A. Job Management Manager Components

A.1. Components

Job Management Manager (the manager) consists of the following files and system components:

A.1.1. Files

For a list of Job Management Manager files supplied in the installation kit, see the *VSI Job Management for OpenVMS Installation Guide*.

A.1.2. System Components

- Job executor (NSCHED.EXE), a detached process that executes jobs
- Job database, a collection of information that describes jobs on a node or cluster of nodes (VSS.DAT)
- DCL interface that includes both management commands and general commands (SCHEDULER\$MANAGER.EXE)
- Motif interface, a workstation approach that provides a graphical interface for the manager (SCHEDULER\$MOTIF.EXE)
- Special Days databases (SCHED\$SD_CLASSES.DAT and SCHED\$SD_RESTRICTIONS.DAT)
- Event-log report utility for logging job events (VSS_REPORTS.EXE)
- Program (RETRY.EXE) gets executed by the manager on regular intervals when an action is being retried. It runs regularly until the actions it is attempting (stored in RETRY.DAT) are successful, or the retry period expires.
- Detached process (SCHED_REMOTE.EXE) that listens for remote Job Management Manager actions, enabling the manager to work over a network. NODENAME_REMOTE.LOG is the output log file for SCHED_REMOTE.EXE.
- Detached process (SCHED\$LISTENER.EXE) that listens for termination messages from the remote Executor
- Database repair utility (DB_UTILITY.EXE)
- Load Balance Groups configuration utility (SCHEDULER\$CONFIG.EXE) that creates, edits, and deletes Load Balance Group definitions in the Load Balance Group database (UJM\$LOAD_BALANCE_GROUPS.DAT) for customized load balancing in an OpenVMS cluster.
- CPU system rating utility (CPU_UTILITY.EXE) that displays the current system rating used in load balancing calculations.

Appendix B. Logical Names, Global Symbols, and Event Logs

B.1. Overview

This appendix provides a list of Job Management Manager (manager) logical names and run time global symbols. The appendix also describes how to write job events to a log file.

B.2. Job Management Manager Logical Names

The following table lists and describes Job Management Manager logical names. Beginning with Release 3.1, the logical names reside in the two files:

NSCHED\$COM:JOB\$MANAGER\$STARTUP_*nodename*.COM

NSCHED\$COM:JOB\$MANAGER\$STARTUP_SYSCLUSTER.COM

The *nodename* in the first filename is replaced by the system node name, and contains system specific logical names. The second file contains cluster-wide logical names.

You can edit these files to modify the values of the logical names. The file NSCHED\$COM:JOB\$MANAGER\$STARTUP_LOCAL.TEMPLATE contains the default values for all logical names, and is provided at installation time.

Logical Name	Description
NSCHED\$	Default set up at installation, based on response from user questions. Points to the directory that contains all of the manager's files.
NSCHED\$ALIAS_GROUPS	Logical name to use to repair stale remote cluster aliases in a job record. See Repairing Job Records Containing Stale Cluster Aliases in this guide.
NSCHED\$BS_MAX_WAIT	Identifies the maximum seconds to wait for "Batch start."
NSCHED\$BS_RETRIES	Identifies the number of times to retry sending "Batch start."
NSCHED\$CHECK_RATIO	Returns the ratio of timer events to check procedure.
NSCHED\$CLEAR_RESTART_PARAM	A job can have a restart parameter set while it is running, so if it fails during execution, the restart parameter can be passed back when it restarts. If this is defined as TRUE, the restart parameter will be cleared when the job completes and the job will start from the beginning the next time it runs. If defined as FALSE the restart parameter will not be cleared when the job completes, thus the next time the job runs it will continue from the saved restart parameter. Default: TRUE.

Logical Name	Description
NSCHED \$CUSTOM_LOAD_BALANCE	If defined to point to a command procedure to execute, the current (default) load balancing will be overridden. The command file will be executed at the time of considering a load balance. The manager expects that command procedure to have defined a logical called NODE_TO_USE as the nodename to use. Default: UNDEFINED.
NSCHED\$DEBUG	Sets debug either on or off. Default: off.
NSCHED\$DEFAULT_JOB_MAX	Defines the maximum number of jobs allowed to run at one time. Default: 6.
NSCHED\$DEFAULT_JOB_PRI	Sets the priority level at which jobs will run. Default: level 4.
NSCHED\$DEFAULT_LOG	Defines the logging events that will be captured. Default: JOB_EVENT, ABNORMAL_EVENT, TERMINATION_EVENT, INTERFACE_EVENT, and ERROR_TRAP.
NSCHED_DEFAULT_SD_ACTION	Defines the system default Special Day Action, which can be SKIP or HOLD. Default: SKIP.
NSCHED\$LBAL\$CPU_WEIGHT	Weighs the CPU rating by multiplying the CPU rating by the translation of this logical. A valid range is from 0.0 to 1.0. Default: 0.5.
NSCHED\$LBAL\$INTERVAL	Determines the number of seconds between data collection. Default: 30.0 seconds.
NSCHED\$LBAL\$MEMORY_WEIGHT	Weighs the memory rating by multiplying the memory rating by the translation of this logical. A valid range is from 0.0 to 1.0. Default: 0.5.
NSCHED\$LBAL\$UPDATE_RATING	When set to ON, decrease the node rating when a job is sent to a node. Cluster logical. Possible values: OFF, ON. Default: OFF.
NSCHED\$LBAL\$WAIT_AVAIL_NODE	If TRUE, puts a job in Job Slot Wait for an available node when a Load Balance Group has no available nodes. If

Logical Name	Description
	FALSE or UNDEFINED, the job is run on the default Job Management Manager node. Default: UNDEFINED.
NSCHED\$LBAL_INSENSITIVE	Defines the percentage difference between node ratings that is considered to be insignificant such that round-robin scheduling can be used. Cluster logical. Default: 0 (disabled).
NSCHED\$LOAD_BALANCE	Turns load balancing on or off. Default: ON.
NSCHED\$LOGFILE	Names the Job Management Manager log file. Default: NSCHED\$:VERMONT_CREAMERY.LOG.
NSCHED\$MAILBOX	System logical that points to the manager's message mailbox. Users should not touch this in any way.
NSCHED\$MESSAGE_RATIO	Returns the ratio of mailboxes messages to timer_event.
NSCHED\$NODEPS	Suppresses the display of job dependencies. Default: FALSE.
NSCHED\$ <i>node</i> _SCHED_PORT	The variable <i>node</i> is the OpenVMS node name defined by SYSS\$NODE in DECnet. If you cannot use the default UDP port 5482 on this system, define this logical to override the default. Default: 5482.
NSCHED\$ <i>remnode</i> _EXEC_PORT	The variable <i>remnode</i> is the name of the target remote node. The node name can be, at most, 13 characters. Use this logical in the case of a remote node that must use a UDP port number other than the default (5481). It may require defining an alias for a remote node so its length is fewer than 14 characters. The variable <i>remnode</i> will be the same as the node specified in the job's REMOTE_NODE characteristic. Default: 5481.
NSCHED\$REMOTE_SUPPORT_ENABLED	The default is defined by the installation, where the installed will be asked questions. If TRUE, remote support will be available. Remote mode jobs and accompanying parameters are supported.
NSCHED\$SILENT	Suppresses the connecting messages to remote nodes.

Logical Name	Description
	Default: FALSE.
NSCHED\$TERM_MAILBOX	System logical that points to the manager's termination message mailbox. Users should not touch this in any way.
NSCHED\$TERMINATION_RATIO	Returns the ratio of termination messages to mailboxes.
NSCHED\$UID	Defines the location of the X-window interface user interface definition (UID) file.
NODE_TO_USE	Defined by users if they have selected custom load balancing. It is a local logical, so often it will not have wide impact.
SCHED\$BATCH_KEEP_LOG	Retains log files from jobs run in batch queues. Default: TRUE.
SCHED\$BATCH_NAME_PREFIX	Defines the prefix added to the batch job name. Default: VJM.
SCHED\$BATCH_PRINT_LOG	Prints log files from jobs run in batch queues. Default: FALSE.
SYS\$NODE	Must be defined as the name of the node before the manager starts.

B.3. Run Time Global Symbols Available to Jobs

The following table shows the run time global symbols created by the manager each time it runs a job. You can reference these global symbols in your command procedures when a job needs information about itself.

For example, a job may want to read its comments (which can be specified in the **RUN** command), its job name within the command procedure, or its job number so that it can change its own operation.

Symbol	Definition
BATCH\$RESTART	The restart value set by the last SCHEDULE SET RESTART command within the job's command procedure if the system failed while the job was running.
NSCHED_COMMAND	The job's main OpenVMS command, created or modified by the job's owner. For example: @SYS\$DEVICE:[DOE.PROJ-C]WKLY-COPY.COM
NSCHED_COMMENT	The job's text comment string, created or modified by the job's owner. This is one way to pass information to a job at run time. For example: Check for weekly lists.

Symbol	Definition
NSCHED_GROUPNAME	The job's group, created or modified by the job's owner. Examples: DAILY, WEEKLY, MONTHLY
NSCHED_JOBNAME	The job's name, created or modified by the job's owner. For example: WEEKLY-REPORT
NSCHED_JOBNUM	The job's unique job number, automatically assigned by the manager when the job was created or copied into the job database. For example: JOB 747
NSCHED_OUTFILE	The file specification for this job's SYS\$OUTPUT and SYS\$ERROR output. For example: SYS\$DEVICE:[DOE.PROJ-C]WKLY-REPORT.LOG
NSCHED\$RETRY_COUNT	The number of times the job has been retried since it last failed.
NSCHED\$RETRY_MAX_ATTEMPTS	The maximum number of retries allowed for the job.
NSCHED_SEVERITY	The severity element of the exit status of the job's prefunction (if any) when checked from the job's main procedure, or of the exit status of the job's main procedure, when checked from the job's postfunction. The value range and purpose are identical to those of the DCL symbol \$SEVERITY. The value consists of the low order 3 bits of the job exit status. As per OpenVMS condition values conventions, even values signal a fatal error, an error, or a warning. Odd values represent either success or informational conditions. See also the symbol NSCHED_STATUS below.
NSCHED_STATUS	The exit status of the job's primary OpenVMS command when last run by the manager. This status is used by the jobs that depend on that job and by the job's postfunction (if any). Refer to Prefunctions and Postfunctions in Chapter 2, "How Job Management Manager Works" for a discussion of prefunctions and postfunctions.
NSCHED_STEP	Symbol defined for a job when it is run (similar to NSCHED_COMMAND or NSCHED_JOBNAME). It has one of the five following values, indicating the current step of job execution from the manager perspective: NSCHED_PRE_START NSCHED_PRE_RUN NSCHED_MAIN_START NSCHED_MAIN_RUN NSCHED_POST_RUN
NSCHED_TYPENAME	The job's type, created or modified by the job's owner. For example:

Symbol	Definition
	PROGRAM, DOCUMENTATION, BACKUP, PROJECT-COSTS
NSCHED_VERSION	Symbol defined for a job when it is run (similar to NSCHED_COMMAND or NSCHED_JOBNAME). It is an 8-character string that indicates the version number of the manager.
\$VSSRESTART	The job's current restart flag, set by the manager. This flag's setting is TRUE only if the job is set for RESTART. Otherwise, the setting is FALSE. Refer to the /RESTART qualifier for the SCHEDULE CREATE command for more details about job restarts.

B.4. Writing Job Events to a Log File

Job Management Manager can write a job event-log file that records when a job starts, completes, or encounters an error. The file also records when other significant system activities occur. Users' actions to control the Job Management Manager or the jobs, or to alter the job database may be logged as well.

The Event Log events are grouped by Event Classes. Select any sub-list of event classes for event logging, among the following available classes:

Class name	Default Selection	Logged Information
JOB_EVENT	X	All job events, specifically job edits and job control events as performed by the Job Management Manager. Job Events are logged as the Job Management Manager actually processes the resulting commands queue or when a job timer triggers.
NORMAL_EVENT		Normal Job Management Manager events, including successful job edits and control events.
ABNORMAL_EVENT	X	Abnormal Job Management Manager events. For example job execution failures and unprivileged access requests denied.
TERMINATION_EVENT	X	Job Management Manager exits.
INTERFACE_EVENT	X	Users' actions through either the command line or the graphical user interface. Interface Events are logged as the user queues requests to the Job Management Manager.
MAILBOX_MESSAGE		Mailbox messages.
BROADCAST_MESSAGE		Broadcast messages to user terminals.
TIMER_SETTING		Job timer settings.
DATABASE_CHECK		Database consistency check (start and finish).
ERROR_TRAP	X	All error trapping within the Job Management Manager and utilities.
STATE_CHANGE		All job state changes.

The following Event Classes are configured for logging by default:

JOB_EVENT, ABNORMAL_EVENT, TERMINATION_EVENT, INTERFACE_EVENT, and ERROR_TRAP.

If you have privileges to use the manager's management commands, you can use logging commands to perform the following functions with this event-log file:

- Control the types of information logged with the **SCHEDULE SET LOGGING** command.
- Display the selection of Event Classes currently active for Job Management Manager event logging with the command **SCHEDULE SHOW LOGGING**.
- Close and rename the event-log file with the **SCHEDULE CLOSE LOG_FILE** command.
- Display detailed Job Management Manager event-log reports with the following command:

```
$ RUN NSCHED$EXE:VSS_REPORTS. EXE
```

The command pair **SET LOGGING /CLUSTER_NODE=node name** and **SHOW LOGGING /CLUSTER_NODE=node name** behaves differently at the DCL level and at the shell because of timing. When you wait 5 seconds between the **SET** and **SHOW** commands, both interfaces behave the same and display correct information. The **SHOW** displays the results of the previous **SET** command. This timing issue occurs because the **SET LOGGING/CLUSTER** command sends a request to NSCHED to change. If you issue the **SHOW LOGGING** command immediately, the shell reacts before the change is made.

The name of the event-log file is controlled by the logical name NSCHED\$LOGFILE. The system manager should define this logical name in the NSCHED\$COM:JOB\$MANAGER\$STARTUP_nodename.COM file. If you do not define this logical name, the default name is NSCHED\$DATA:VERMONT_CREAMERY.LOG.

The default list of Event Classes is governed by the logical name NSCHED\$DEFAULT_LOG. See that logical name definition in NSCHED\$COM:JOB\$MANAGER\$STARTUP_LOCAL.TEMPLATE for details.

For more details on the above commands, see the individual command descriptions in the *VSI Job Management for OpenVMS Command Reference Guide*.

Appendix C. Customizing Load Balancing

C.1. Overview

This appendix provides information on customizing your own load balancing algorithm.

C.2. Customizing Load Balancing

To use fully customize load balancing, you need to define the logical NSCHED \$CUSTOM_LOAD_BALANCE to the value TRUE. Then Job Management Manager (the manager) spawns the command procedure NSCHED\$COM:LOAD_BALANCE.COM each time a job is to be run. The job runs on the node pointed to by the logical NODE_TO_USE that is defined in the load balance command procedure.

The manager provides two load-balance command procedures:

- The default algorithm is implemented in the LOAD_BALANCE.COM file. This algorithm relies on a program that calculates the relative load rating of each node in an OpenVMS cluster, based on memory and CPU ratings.
- An alternative algorithm is implemented in the file LBAL_ROUND_ROBIN.COM. This algorithm uses simple round-robin load balancing.

You can set the NSCHED\$CUSTOM_LOAD_BALANCE logical in the manager's startup procedure to TRUE or FALSE. If TRUE, it spawns the file NSCHED\$:LOAD_BALANCE.COM to perform the load balancing algorithm. The existence of this file is the responsibility of the system managers.

SCHEDULER\$LBAL and SCHEDULER\$LBAL_MGMT, present in Job Management Manager V2.0 and earlier, have been removed from the product.

SCHED SHOW LOAD_BALANCE [/NODE=node, /ALL] shows full rating information.

SCHED SHOW STATUS now contains the rating in place of the LBAL string. When load balancing is enabled, the "sched_flags" field in the return buffer from VSS\$SHOW_SCHED returns the actual rating as a numeric string rather than LBAL.

VSS\$GET_BEST_NODE (new function) returns the name of the node to select for load balancing.

VSS\$GET_LBAL_INFO (new function) is called with a nodename and returns load rating components into a 16-byte buffer passed by reference.

The file LOAD_BALANCE.COM supports the ability to restrict a job or a group of jobs to run on a list of nodes, instead of to a single node.

The SCHED\$GET_BEST_NODE.EXE program defines the logical NODE_TO_USE to be the best rated node in the cluster. It is possible to pass this program a parameter that is a list of nodes, separated by spaces, from which the program will choose the best-rated node.

Two system logical names can be defined to achieve this behavior:

```
DEF/SYS SCHED$LBAL_GROUP_groupname "node1 node2 node3"
```

This allows the jobs with the given group name to be load balanced on one of the three listed nodes.

```
DEF/SYS SCHED$LBAL_JOBNUM_jobnumber "node2"
```

This overrides the job group restriction above and further restricts the given job to the given node.

If " " is given as the node parameter list, all available nodes in the cluster will be used.

Note

If the LOAD_BALANCE.COM procedure cannot identify a node on which to run the job (specified Job Management Manager does not exist), the job it is attempting to run is put in HOLD state and you are notified as appropriate by mail, OPCOM, and terminal notify.

C.3. Restricting a Job to Run on a Specific Node

To restrict a job to run on a specific node when you use fully customized load balancing, follow these steps:

1. Define the logical name NSCHED\$CUSTOM_LOAD_BALANCE as TRUE:

```
$DEFINE/SYS/EXEC NSCHED$CUSTOM_LOAD_BALANCE "TRUE"
```

2. Define the logical name SCHED\$LBAL_JOBNUM_### to point to the node where you want the job to run:

```
$DEFINE/SYSTEM/EXEC SCHED$LBAL_JOBNUM_3 "nodename"
```

Appendix D. Using Extended Job Fields

To take advantage of extended field lengths for certain job fields, you can enter fields that exceed the DCL token length of 255 characters. Follow the best practices contained in this appendix when specifying job fields that are longer than a standard OpenVMS command token.

D.1. General Best Practices

The following list provides best practices to specify job fields that are longer than a standard OpenVMS command token.

- Use the DCL command line when creating or modifying few jobs that have no data fields longer than 255 characters.
- Use the Motif GUI when creating or modifying few jobs that have one or more data fields greater than 255 characters.
- Use prompting when creating or modifying a job from the DCL command line that has at least one data field greater than 255 characters. Depending on the fields used, you can initiate either [full prompting or brief prompting](#).
- Design a [command script](#) to generate the jobs when creating or modifying many jobs.

D.2. Prompting Best Practices

Use prompting when creating or modifying a job from the DCL command line that has at least one data field greater than 255 characters. Depending on the fields used, you can initiate either full prompting or brief prompting. You can initiate either full or brief prompting to modify more than one field at the same time.

- Full prompting is initiated using the **/PROMPT** qualifier for the **CREATE**, **MODIFY**, and **COPY** commands.
- Brief prompting is initiated when no parameters or qualifiers are entered for the **CREATE**, **COPY**, or **MODIFY** commands on the command line.

Note

For more information about how each command handles prompting, see the commands in the *VSI Job Management for OpenVMS Command Reference Guide*.

When you use prompting, note the following conditions:

- You cannot enter more than 255 characters per line.
- You can enter the continuation character (a single dash "-") before reaching the 255 character limit. The dash can be character 255.

Depending on the field you used, you can initiate either full or brief prompting to modify more than one field at the same time.

- Use the following commands to specify a `VMS_COMMAND` field greater than 255 characters, and to initiate brief prompting:
 - **CREATE** without specifying the VMS command as the parameter
 - **COPY** without specifying the `/VMS_COMMAND` qualifier
 - **MODIFY** without specifying any qualifier that modifies the job fields
- Use the following commands to specify an output file field greater than 255 characters. The commands initiate brief prompting that includes the `/OUTPUT` qualifier.
 - **CREATE** or **COPY** without specifying the `/OUTPUT` qualifier
 - **MODIFY** without specifying any qualifier that modifies the job fields
- Use the following commands to specify a prefunction or postfunction field greater than 255 characters, and to initiate full prompting. The prefunction and postfunction qualifiers are not included with brief prompting.
 - `CREATE/PROMPT`
 - `COPY/PROMPT`
 - `MODIFY/PROMPT`

D.3. Script Best Practices

Design a command script to generate the jobs when creating or modifying many jobs. The following list provides script best practices for specifying extended field lengths.

- You can create the script from the beginning.
- You can use the **SCRIPT JOB** command to script existing jobs and then edit the result to meet your needs. The output from the **SCRIPT JOB** command is designed to recreate jobs, so it uses the **CREATE** command exclusively.
- If you want to copy or modify jobs in the scripted database, change the **CREATE** command to **COPY** or **MODIFY** commands with other required edits.

Appendix E. Troubleshooting and FAQs

This appendix provides some troubleshooting tips for Job Management Manager (the manager) and Job Management Agent (the agent) as well as some answers to frequently asked questions.

E.1. Troubleshooting the Manager

The following tables provide tips for diagnosing and fixing some common Job Management Manager issues:

E.1.1. Next Run Time Is NEVER

Possible Causes	Possible Solutions
No days specified	Modify job to have some days specified
No interval specified	Modify job to have an interval, and run the job
No start time specified for new job	Modify job to have a next start time
Job was modified /NEXT=NEVER	Modify job to have a next start time
Job was restricted to a Special Day Class with no days in the future	Modify next start time or Special Day Class

E.1.2. Next Run Time Is Wrong

Possible Causes	Possible Solutions
Encountered Special Day Restrictions	Relax or change job's Special Day Restrictions, or modify Special Day Class
Fiscal calendar not as expected	See documentation for SCHEDULE SHOW FISCAL_YEAR and SCHEDULE SET FISCAL_YEAR for fiscal year characteristics
USE_NEXT_TIME in effect	Verify that /USE_NEXT_TIME is appropriate for the job
Job was modified by someone	N/A
Job already ran, or did not run when expected	N/A

E.1.3. Job Did Not Run or Did Not Run at Specified Time

Possible Causes	Possible Solutions
Incorrect interval	Fix interval

Possible Causes	Possible Solutions
Output file disk full	Check for insufficient disk space
Output file disk not accessible	Check for or disk not mounted or in mount verify timeout
Node restriction	Relax or modify node restriction
Special Day Restrictions	Relax or modify Special Day Restrictions
Dependencies not satisfied ([])	1. Check parent jobs for successful completion 2. Check for new or changed remote cluster aliases in job records. See Repairing Job Records Containing Stale Cluster Aliases in this guide.
Dependencies failed	Check parent jobs for reasons for failure
Remote dependency or network down	Check why network connection is not available
Dependency did not run	Check why parent job did not run
Dependencies entered, but interval not entered	Check interval, or set interval to 0
One or more dependencies satisfied unexpectedly	Check for manual or out-of-order runs of parent jobs
On HOLD	Release job
No job slots available	Wait, increase /MAX_JOBS
Incorrect next run time	Modify job's next run time
No privilege to run	Check ownership, privileges, names, identifiers
Job was not scheduled for expected time	No action required
Load Balance Group assigned to the job is empty	Add one or more nodes to the Load Balance Group or remove the Load Balance Group from the job

E.1.4. Job Runs Unexpectedly

Possible Causes	Possible Solutions
Someone ran the job manually	No action required
Dependencies completed or were overridden	No action required
The manager just became available	No action required
Specific node just became available	No action required
Job was modified to change restrictions	No action required
Job is an SJOB or TJOB, and ran because another job stalled or timed out	No action required

E.1.5. Notification of Special Days Restriction on Job and It Still Runs

Possible Cause	Possible Solution
The notification is for the NEXT scheduled time, not for THIS one	N/A

E.1.6. Cannot Access Job by Name

Possible Cause	Possible Solution
Name misspelled	Use wildcards with SHOW to find correct job name
Different user name	Use the /USER qualifier with SHOW to find proper job
No access privilege	Check ownership, privileges, names, identifiers
Job on different machine	Use the /SERVER qualifier
Job has no name	Modify job to have desired name

E.1.7. Job Fails Immediately at Run Time

Possible Cause	Possible Solution
Improper setup of files in NSCHED	Have system manager check NSCHED\$:*.* against documentation for existence, protection, and installation of files
Output filename incorrect	Check /OUTPUT file specification validity
No privilege for access to files	Check filespec, user, ACLs, directory protection, etc.
Improper use of logical names	Cannot use non-system logical names, defaults to SYSMGR
Command file missing, or invalid VMS_COMMAND	Check VMS_COMMAND , replace if invalid
No such user	Change username on job
User account unavailable (DISUSER...)	Check with system manager

E.1.8. Job is Missing

Possible Causes	Possible Solutions
Name was changed	Try using job number, change name again, use wildcards
Username was changed	Use /USER= or " <i>name=user</i> " format for access
Job had /NORETAIN set	Run VSS_REPORTS to check for deletion
Someone deleted job	Run VSS_REPORTS to check for deletion
VSS.DAT was restored from a backup file	Old database from before job was created --CREATE job again

Possible Causes	Possible Solutions
Looking on wrong machine	Check other machines, or use of /SERVER
NSCHED\$DATA logical pointing to wrong database	See system manager

E.1.9. Output File Is Not Being Written

Possible Causes	Possible Solutions
Job is being written to wrong location	Check file spec, directory defaults to SYS\$MANAGER
Job did not run	Check job's last run time
Not checking output file specified for job	MODIFY JOB /OUTPUT=filename.DAT
Invalid output filespec	Modify job to have proper output filename
Disk space or quota exceeded	See system manager

E.1.10. No Notification Upon Completion

Possible Causes	Possible Solutions
Job was run under different username	N/A -- notification is sent to user who runs the job
Notification not set up	Use /NOTIFY qualifier
User not logged in to get notification	N/A
Job did not run	N/A
Job ran on a different machine	N/A

E.1.11. No Mail Upon Completion

Possible Causes	Possible Solutions
/SEND not properly set	Use /ALL or check status of last exit
/MAIL_ADDRESS not properly set	Make sure a valid user or distribution file is specified
Disk space or quota exceeded, DISMAIL set, or other system reason	See system manager
Job did not run	N/A

E.1.12. No OPCOM Upon Completion

Possible Cause	Possible Solution
The notification is for the NEXT scheduled time, not for THIS one	N/A

E.1.13. Job Aborts

Possible Causes	Possible Solutions
BATCH mode job exceeded maximum CPU time	Raise /CPU_LIMIT on job

Possible Causes	Possible Solutions
Manual intervention (SCHED ABORT)	N/A
System crash	N/A
OpenVMS process terminated by system or user	N/A
Job completed without the manager running to notify	N/A

E.1.14. Job Successful But Shows Termination Status as Failure

Possible Causes	Possible Solutions
Command file resets termination status	Check command procedure for proper exit status
Postfunction resets termination status	Check command procedure for proper exit status
Program and application use different definition of SUCCESS/FAILURE	N/A
Job did not run, or ran again since it was viewed; looking at wrong status	Check last completion time

E.1.15. Job Management Manager Is Slow in Responding

Possible Causes	Possible Solutions
System resources fully used	N/A
<i>nodename</i> .LOG > 500 blocks	Use SCHEDULE SET DEBUG OFF or SCHEDULE STOP
History log > 2500 blocks	Use SCHEDULE CLOSE LOG [/SUMMARIZE]
Database fragmented; deletes > 200	Use SCHEDULE OPTIMIZE DATABASE to compress database
Many interface requests at the same time	N/A
Job Management Manager default transition	N/A
New scheduler joining cluster	N/A
NSCHED priority has been lowered	See system manager
NULL keys are not present in the jobs database	Shutdown JM Manager cluster wide and execute the DCL command \$ CONVERT/FDL=NSCHED\$DATA:VSS_STANDARD.FDL NSCHED

E.1.16. Load Balancing Is Not Working

Possible Causes	Possible Solutions
Job has a node restriction	Use SCHEDULE MODIFY JOB /NOCLUSTER_NODE

Possible Causes	Possible Solutions
Load balancing turned off	Use SCHEDULE SET LOAD_BALANCE ON
Weighting factors set to 0	Check logicals and see system manager
Only one node has reasonable rating	Check ratings using SCHEDULE SHOW STATUS
Load Balance Group assigned to the job is empty	Add one or more nodes to the Load Balance Group or remove the Load Balance Group from the job
NSCHED\$LBAL \$WAIT_AVAIL_NODE setting is incorrect	Check the logical and see system manager.

E.1.17. Jobs Going into Waiting State

Possible Cause	Possible Solution
There are MAX_JOBS jobs running on the node on which this job is to run	Check using SCHEDULE SHOW STATUS and reset using SCHEDULE SET MAX_JOBS

E.1.18. Jobs Unexpectedly on HOLD

Possible Cause	Possible Solution
Job fails, has /NORESTART set	Check why job failed, check restart setting of job
HOLD requested by user	N/A
HOLD set as Special Day Action and job met HOLD criteria	N/A

E.1.19. SCHED_REMOTE Will Not Start

Possible Cause	Possible Solution
Account does not have DEFAULT privileges to run or to access NSCHED\$:SCHED_REMOTE.COM	See system manager, start the manager from SYSTEM account

E.1.20. SCHED\$LISTENER Will Not Start

Possible Cause	Possible Solution
Account does not have privileges to run or access NSCHED\$:SCHED\$LISTENER.COM	See system manager, start the manager from SYSTEM account
Remote support not enabled on machine	See system manager, check documentation

E.1.21. Job Status = NOT RUNNING

Possible Cause	Possible Solution
The manager is running slowly and therefore not reading its mailbox quickly enough to update the database information	Check number of deleted jobs, size of the history log, size of the DEBUG log if turned on

E.1.22. Job in BATCH Mode Does Not Show in SCHEDULE SHOW JOB/BRIEF

Possible Cause	Possible Solution
Job is owned by a different user	SCHEDULE SHOW JOB/FULL command will give a full display of the database profile for this job so user can verify information

E.1.23. DECwindows User Interface Results in Xtoolkit Error: Can't Open Display

Possible Cause	Possible Solution
System is not authorized to display the application windows	Use DCL command SET DISPLAY/CREATE/NODE=<i>nodename</i> , or add NODENAME::USERNAME in Session Manager Security Options dialog box

E.1.24. NSCHED Process Goes into RWMBX State

Possible Cause	Possible Solution
Insufficient quota	Try increasing BYTLM and/or PQL BYTLM

The following conditions determine the run time for a job: Jobs run when the specified scheduler is accessible AND

- scheduled time is in the past
- all dependencies have been met
- there are available job slots

OR

- there is a manual request to run the job and
- there are available job slots

OR

- the job is an SJOB or TJOB and
- the related job is stalled or timed out and
- there are available job slots

E.1.25. Recreating Deleted Database

If the jobs database VSS.DAT was deleted for any reason, you can recreate a new database with the optimal key attributes.

Follow these steps:

- 1. Start the Job Management Manager and allow it to create a new empty database.

2. Shutdown the product cluster wide using the command:

```
$ SCHEDULE STOP/ALL
```

3. Run the convert utility on the new database:

```
$ CONVERT/FDL=NSCHED$DATA:VSS_STANDARD.FDL NSCHED$DATA:VSS.DAT NSCHED$DATA:V
```

E.1.26. Repairing Job Records Containing Stale Cluster Aliases

If a job has a remote dependency specified using an alias of a remote cluster and that remote cluster alias is changed, the job can get stuck in DepWait state. This might also occur if groups of nodes use different cluster aliases within the same cluster.

Other symptoms are:

- Job dependencies cannot be shown in **SCHEDULE SHOW JOB/FULL** output and in the Motif GUI.
- RETRY.LOG is filled with multiple attempts to retry remote connections.
- Removing a dependency or deleting a job leaves stale dependency records in DEPENDENCY.DAT

This happens because dependency records stored in the database are not updated when a remote cluster alias changes. Therefore, any connection to the stale cluster alias fails and job dependencies cannot be satisfied.

Use the logical name NSCHED\$ALIAS_GROUPS to tell Job Management Manager about cluster aliases which have been added or changed for remote clusters or nodes, when using inter-cluster dependencies. The logical name allows Job Management Manager to resolve mismatches in the database, update job records if necessary, and proceed with job dependencies.

E.1.26.1. Using NSCHED\$ALIAS_GROUPS to repair cluster aliases

Note

The word "name" is used to refer to either a cluster alias name or a single node name.

Note

Only define NSCHED\$ALIAS_GROUPS if cluster alias changes are made or planned. The product works more efficiently if the logical name is not defined.

Define the logical name on remote clusters and/or nodes which have dependency relations with the cluster where the alias is being changed. It does not need to be defined in the cluster where alias name gets changed. In other words, the logical name should not contain any local cluster alias or node names. Dependency records are updated when a job attempts to run and checks its dependencies and when "dependency job completed" notifications are sent.

If a cluster alias has been changed and the old alias is not accessible in the network, define the logical name equivalence string to include both old and new alias in order for Job Management Manager to resolve the dependencies.

If a cluster uses multiple aliases, define the logical name equivalence string to contain all of them.

Individual cluster node names are also allowed in the equivalence name. However it only makes sense to include a node name if a job dependency was set using that node name.

The logical name can be defined as a search list where each equivalence name represents comma separated names for one cluster. This way a single logical name can be used to resolve names for multiple clusters.

Warning

Each equivalence name may only contain names for the same cluster. Putting names for different clusters into the same equivalence name is invalid and might lead to unpredicted results.

When the logical name is properly defined, Job Management Manager resolves conflicts between names in job records and actual names in the network. It then updates these names in the job database automatically.

If a name that is defined in the job record cannot be connected to, Job Management Manager attempts to connect to alternate names defined in the logical name until a successful connection is made, after which it updates the job record with the successfully connected name.

If a dependency job completion notification is received from a node whose name differs from the one defined in the job record (and both names are defined in the same equivalence name), Job Management Manager updates the dependency record using the name the notification came from.

The recommended order of names in each equivalence name is as follows:

1. Old (inactive) name
2. Active cluster alias
3. Individual node name

The logical name can be defined in the LNM\$SYSCUSTER_TABLE to be used by all cluster nodes. Alternately, it can be defined in LNM\$SYSTEM_TABLE on nodes which need job dependency node name resolution.

E.1.26.2. Sample scenarios

A single alias or node name length cannot exceed 6 characters.

The following sample scenarios illustrate the use of the logical name NSCHED\$ALIAS_GROUP to resolve cluster alias problems. The following assumptions are used for all examples:

Begin with three Job Management Manager configurations:

- A two node cluster with nodes ALPHA1 and ALPHA2 using cluster alias ALPHAS.
- A two node cluster with nodes EARTH and MARS using cluster alias PLANET.
- A two node cluster with nodes PRD1 and PRD2 not using any cluster alias.

Begin with these job dependencies:

- Job ALPHAS::ALP1 is dependent on job PLANET::E1
- Job ALPHAS::ALP2 is dependent on job PRD1::PROD1 which is set to run on PRD1 only.

- Job PLANET::PL1 is dependent on job ALPHAS::ALP3
- Job PLANET::PL2 is dependent on job PRD2::PROD2 which is set to run on PRD2 only.
- Job PRD1/PRD2::PROD3 is dependent on job ALPHAS::ALP4
- Job PRD1/PRD2::PROD4 is dependent on job PLANET::M1

The logical name NSCHED\$ALIAS_GROUPS is not defined, everything works as expected.

Scenario 1

Two nodes are added to the ALPHAS cluster. New nodes are IA64 architecture, therefore it is decided that they use a different cluster alias, IA64S. Job Management Manager has been installed on the new nodes and all four cluster nodes use the same job database. The jobs ALP1, ALP2, ALP3 and ALP4 are allowed to run on any node in the cluster.

Implications: When jobs ALP3 and ALP4 run and complete on one of new IA64 nodes, their dependents PLANET::PL1 and PRD1/PRD2::PROD3 get stuck in DepWait state and never run. Additionally, if the two Alpha nodes ALPHA1 and ALPHA2 are down but the IA64 nodes are running, jobs ALP1 and ALP2 will not run when their dependencies PLANET::E1 and PRD1::PROD1 are complete.

Solution:

Define the logical name NSCHED\$ALIAS_GROUPS as follows on both the PLANET and PRD1/PRD2 clusters:

```
$ DEFINE/TABLE=LNMS$SYSCLUSTER_TABLE NSCHED$ALIAS_GROUPS "ALPHAS, IA64S"
```

Results after defining the logical name:

When jobs ALP3 and ALP4 run and complete on one of new IA64 nodes, their dependent jobs PLANET::PL1 and PRD1/PRD2::PROD3 start as expected and their dependency records are changed as follows:

Job PLANET::PL1 depends on IA64S::ALP3

Job PRD1/PRD2::PROD3 depends on IA64S::ALP4

If nodes ALPHA1 and ALPHA2 are down, jobs ALP1 and ALP2 start on an IA64 node and their dependency records are changed as follows:

Job IA64S::ALP1 is dependent on job PLANET::E1 Job IA64S::ALP2 is dependent on job PRD1::PROD1

Scenario 2

The cluster alias PLANET is changed to SPHERE. The old cluster alias PLANET is not accessible over the network.

Implications:

The following jobs might get stuck in DepWait state:

```
ALPHAS::ALP1
PRD1/PRD2::PROD4
PLANET::PL1
PLANET::PL2
```

Solution:

Define the logical name NSCHED\$ALIAS_GROUPS as follows on the ALPHAS and PRD1/PRD2 clusters:

```
$ DEFINE/TABLE=LNMS$SYSCUSTER_TABLE NSCHED$ALIAS_GROUPS "PLANET, SPHERE"
```

Results after defining the logical name:

All jobs run as expected. Dependency records are updated as follows: Job ALPHAS::ALP1 is dependent on job SPHERE::E1

Job SPHERE::PL1 is dependent on job ALPHAS::ALP3 Job SPHERE::PL2 is dependent on job PRD2::PROD2

Job PRD1/PRD2::PROD4 is dependent on job SPHERE::M1

Scenario 3

A new cluster alias PRODS is introduced on PRD1/PRD2 cluster. The jobs PROD1 and PROD2 are modified to run on any cluster node.

Implications:

When the job PROD1 completes on node PRD2, its dependent job ALPHAS::ALP2 won't start. When the job PROD2 completes on node PRD1, its dependent job PLANET::PL2 won't start. If either of the PRD1/PRD2 nodes is down, jobs PROD3 and PROD4 might get stuck in DepWait state.

Solution:

Define the logical name NSCHED\$ALIAS_GROUPS as follows on the ALPHAS and PLANET clusters:

```
$ DEFINE/TABLE=LNMS$SYSCUSTER_TABLE NSCHED$ALIAS_GROUPS "PRODS, PRD1, PRD2"
```

Results of the above logical name definition:

All jobs run as expected. The dependency records are updated as follows: Job ALPHAS::ALP2 is dependent on job PRODS::PROD1

Job PLANET::PL2 is dependent on job PRODS::PROD2 Job PRODS::PROD3 is dependent on job ALPHAS::ALP4 Job PRODS::PROD4 is dependent on job PLANET::M1

E.2. Troubleshooting Job Management Agent

The following sections provide tips for diagnosing and fixing some common Job Management Agent issues.

E.2.1. Job Management Manager Runs a Job on an Agent

When the manager tries to run a job on an agent, the following two types of errors can occur:

- Error occurs preventing the job from starting.
- Error occurs during the job run

When an error occurs that prevents the job from starting, the manager never sends a message that the job started. However, the manager sends a job completion message that includes one of the error codes listed in the section Agent Error Codes.

E.2.2. Missing Information in the .ERR File

When it runs, the agent produces an .ERR file; however, you may not find the information you need. To find the information, you must look at the agent's output file: `SYSS$SYSTEM:NSCHED$AGENT.LOG`.

To turn on debugging in the agent, follow these steps:

1. Open the following file:

```
SYSS$SYSTEM:NSCHED$AGENT_RUN.COM
```

2. Find the following code:

```
$!AGENT_PORT := 0
$!DEFAULT_DIRECTORY := 0
$!DEBUG_LEVEL := 0
$!RUN_DETACHED = 1
$!
$SCHED$AGENT !'AGENT_PORT'
'DEFAULT_DIRECTORY' 'DEBUG_LEVEL'
'RUN_DETACHED'
```

3. Remove the exclamation points to uncomment the four parameters and the NSCHED\$AGENT line. Your code should look like the following:

```
$AGENT_PORT := 0
$DEFAULT_DIRECTORY := 0
$DEBUG_LEVEL := 0
$RUN_DETACHED = 1
$!
$SCHED$AGENT 'AGENT_PORT'
'DEFAULT_DIRECTORY' 'DEBUG_LEVEL'
'RUN_DETACHED'
```

4. Set DEBUG_LEVEL to 5.

Note

You must uncomment all parameters because the parameters are position-dependent.

5. Close the file.

E.2.3. Proxy Validation Error When Connecting to a Remote Node

If you can connect to a remote node, but a proxy validation error occurs, the name of the manager node may not be defined in the agent's hosts database. When this is true, the agent cannot translate the Job Management Manager IP address into a name.

To define the manager node in the agent node's UCX database, use the following command:

```
UCX> SET HOST UJM_Manager_node_name
/ADDRESS=UJM_Manager_node_ip_address
```

For a node running TCPWARE, you must edit TCPWARE:HOSTS. For MULTINET, you must edit MULTINET:HOSTS.LOCAL.

E.2.4. SYS\$NODE Incorrectly Defined

Agent errors can occur if the logical name sys\$node is incorrectly defined. Check sys\$node on the agent node to make sure the logical name:

- is defined.
- ends in two colons (::).
- does not start with an underscore (_).
- is the same as the IP nodename.

If the DECnet nodename and the IP nodename differ, edit SYS\$SYSTEM:NSCHED \$AGENT_RUN.COM to define sys\$node as a process logical name with the IP nodename for its value.

E.2.5. Agent Error Codes

The following table lists error codes you may receive when an agent runs:

Symbol Name	Hex	Decimal Severity Value	Value	Description
NSCHED \$_EXENORMAL	S	0C9692C1	211194561	Normal successful completion
NSCHED \$_EXEINIFAI	F	0C968CE4	211193060	Failed to initialize communication
NSCHED \$_EXEJOBRUN	E	0C968A5A	211192410	Job is already running
NSCHED \$_EXENOTINET	E	0C968A62	211192418	Packet is not INET domain
NSCHED \$_EXERESPORT	E	0C968A6A	211192426	Packet is not from a reserved port
NSCHED \$_EXEBADNODE	E	0C968A72	211192434	The manager node is not known to the agent
NSCHED \$_EXEBADPACK	E	0C968A82	211192450	Unknown agent username
NSCHED \$_EXEBADHOME	E	0C968A8A	211192458	Invalid home directory for agent username
NSCHED \$_EXEBADPROX	E	0C968A92	211192466	Invalid proxy
NSCHED \$_EXENOLOGIN	E	0C968A9A	211192474	Logins are disabled

NSCHED \$_EXEBADPRO	E	0C968AA2	211192482	Mismatched protocol
NSCHED \$_EXEPADVER	E	0C968AAA	211192490	Mismatched Job Management Manager/ Agent version
NSCHED \$_EXEUNKPACK	E	0C968AB2	211192498	Unknown packet type
NSCHED \$_EXEOVERMAX	E	0C968ABA	211192506	Max job count
NSCHED \$_EXEUNKERR	E	0C968AC2	211192514	Unknown error on agent
NSCHED \$_EXEBADSEND	E	0C968ACA	211192522	Unable to send packet to agent
NSCHED \$_EXEBADRECV	E	0C968AD2	211192530	Failed to receive confirmation packet from agents
NSCHED \$_EXEPKTUNLD	E	0C968ADA	211192538	Unable to unload packet
NSCHED \$_EXEPKTLOAD	E	0C968AE2	211192546	Unable to load packet
NSCHED \$_EXENOJOB	E	0C968AEA	21112554	No such job on agent
NSCHED \$_EXENOTERM	E	0C968AF2	211192570	Invalid string descriptor
NSCHED \$_EXENOSHUT	E	0C968B02	211192578	Unable to shutdown agent
NSCHED \$_EXEJOBABORT	E	0C9681EC	211190252	Job was aborted
NSCHED \$_JOB RUNERR	E	0C9681E4	211190244	Error running job

E.3. Frequently Asked Questions

This section provides answers to some of the most frequently asked questions about Job Management Manager.

E.3.1. General Job Management Manager Questions

How can I see what version of Job Management Manager I am running and other information about the manager?

Use the following command:

```
SCHEDULE> SHOW STATUS
```

My account UIC has been changed. How do I change or modify the UIC in the manager's database?

1. From an account with the CMKRNL privilege (such as the SYSTEM account), enter a command similar to the following:

```
SCHEDULE> MODIFY *=SMITH/USERNAME=SYSTEM
```

This command line changes ownership of all jobs owned by username Smith to username SYSTEM. Carefully note the list of job numbers that are changed, because SYSTEM may already own a number of jobs. If SYSTEM already has a job with the same name as Smith, Smith will also have to change the job name because job names must be unique within an OpenVMS username. Using a GROUP name is a way of keeping the job separated from jobs that belong to the SYSTEM account.

2. Enter the following command to display job names and numbers, and be prompted if you want to change the username for each. Answer yes *only* if the job was originally your job. Those jobs will then have your new UIC.

```
SCHEDULE> MODIFY *=SYSTEM/USERNAME=SMITH/CONFIRM
```

I have a job that runs successfully in batch, but fails in the manager with the following error. What's the problem?

```
RMS-E-DEV, Error in device name or inappropriate device type for operation
```

If you have specified an output file, check to ensure that the file specification is defined at the system level. Output file specifications cannot reference a file using logicals that were set up at the time of login. A logical used to specify an output file must be a system logical.

For example, an output file named SYSS\$LOGIN:my_job.LOG will fail because it is not defined at the time the process was created. An output file named SYSS\$MANAGER:my_job.LOG will work successfully because SYSS\$MANAGER is a system-wide logical.

Why does my stall job (SJOB) run more than once?

If a job is stalled (not started by the time it should have), the designated notification occurs. If the manager "looks" at that job (anytime a job is modified), the manager discovers that the job is stalled and sends the notification again, possibly running any SJOBs again.

E.3.2. Setting Up Jobs

How can I specify a job to run on all nodes in a cluster at the same time?

First, use the **SCHEDULE CREATE** command to create a new job to run on one node. Then use the **SCHEDULE COPY** command to create additional clones of that job for each CPU. If you specify a **/GROUP** qualifier when you create the job, you can copy the group and run all jobs in the group at the same time. For example:

```
SCHEDULE> CREATE /GROUP=group-name /CLUSTER_NODE=node1/PROMPT
(You will be prompted for additional information)
SCHEDULE> COPY job-number /CLUSTER_NODE=node2
SCHEDULE> COPY job-number /CLUSTER_NODE=node3
SCHEDULE> COPY job-number /CLUSTER_NODE=node4
:
:
SCHEDULE> RUN /GROUP=multijob
SCHEDULE> SHOW JOB/GROUP=multijob
```

How do I set up a job to run every six hours, Monday through Friday? Use the following command to create the job:

```
SCHEDULE> CREATE "vms-command"/INTERVAL="+0 6"/DAYS=(ALL,NOSAT,NOSUN)
```

If you are running in an OpenVMS Cluster, you can add the **/CLUSTER** qualifier to control on which OpenVMS Cluster node the command executes.

I have scheduled a job that requires mail notification. The job runs okay, but no mail is received. What is the problem?

Ensure that the ENQLM quota of the account that the manager runs on is at least 300. How do I stop a continuous job?

Enter the following command:

```
SCHEDULE> MOD job-name/NEXT=NEVER/NOINT
```

or

```
SCHEDULE> HOLD job-name
```

What happens when daylight savings time ends? Will a job that runs at 23:00 run again when the time falls back an hour?

Setting the clock back an hour should not affect scheduled jobs that have an interval greater than or equal to HOURLY. The reason is that the manager calculates the next time a job will run as soon as the job is ready to run. Moving the clocks ahead an hour also has no effect on jobs, because a job will run if its scheduled time is earlier than the current time.

E.3.2.1. Examples

- A job is scheduled to run every night at 23:00. At 23:00 tonight, just before it starts running, the job sets its next time to run for 23:00 tomorrow night.
- Assume you set back the clock one hour at 23:00. This will not affect the job, because its scheduled time is 23:00 tomorrow night. It will not run twice in the same night.
- At 2:00 A.M. you set your clock ahead to 4:00 A.M. Jobs scheduled to run at 3:00

A.M. start running immediately, because 3:00 A.M. is earlier than 4:00 A.M.

I have a job that must run on Wednesday of the third week of the fiscal month, except when the fiscal month has five weeks. In that case, the job should run on the Wednesday of the fourth week. How do I specify this?

Use the following interval:

```
/INTERVAL="F M -D11 HH:MM"/DAYS=(WED,NONE)
```

This interval specifies the last day of the fiscal month minus 11 days, which is the second-to-last Wednesday of the fiscal month. When you need to specify this type of interval, use your fiscal calendar to calculate the number of days to specify.

I have a complex job that needs to restart if the system fails or if the job fails because of an error at the step last marked with **SCHEDULE SET RESTART_VALUE STEP_NAME**. How do I restart at this step if the job fails because of an error?

Start every job with the following line:

```
$ IF P1 .NES. "" THEN GOTO P1
```

When you need to restart, use the following command:

```
$ SCHEDULE RUN my_job/PARAM=step_01
```

Step_01 is the step where the job should restart.

What should I do when jobs do not start at their scheduled time?

Stop the manager. Turn on **DEBUG** on the manager and view the log. (**DEBUG** can be enabled in the manager startup procedure, or by the **SCHEDULE SET DEBUG ON** command.)

DEBUG creates a file named **NSCHED\$:nodename.LOG**. The log file contains useful information about what is happening with the manager. Also, use the **SCHEDULE SHOW STATUS** to view other useful information.

E.3.3. Mail and Broadcast Messages

How do I turn off the manager's broadcast messages?

When you create or modify a job, use the **/PROMPT** qualifier and answer No to the Notify prompt. (The default answer is Yes, notify on completion.)

Example:

```
SCHEDULE> CREATE/PROMPT
_VMS Command: @dual:[example.notes]nightly_notes.com
_Job Name [None] : extract_notes
_Group [None] : notes_group
_Type [None] : nightly
_Stall Notify [None] :
_Stall Time Job [None] :
_Schedule Interval [None] : d 03:00
_Days [ALL]: (all,nosat,nosun)
_Special Day Restrictions [None] :
_Start Time [15-JAN-2012 03:00:00.00] :
_Output file [None] :dual:[example.notes]nightly_log
_Mode (Detached, Batch, Remote [DETACHED] :)
_Cluster Node [Default] :
_Retain (Success, Error, None or All) [ALL]:
_Restart (Yes, No [Yes]:) N
_Retry (Yes, No [No]:) N
_Notify (Yes, No [No]:
_Send-Mail (All, Error, Success or None) [NONE]:)error
_Send MAIL on Special Day Action (Yes, No [No]:
_Use_next_time (Yes, No) [No]:
OPCOM (Yes, No) [No] :
_Mail_Address [EXAMPLE] :
_User_Name [EXAMPLE] :
_Run Priority [Default] :
_Pre Function [None] :
_Post Function [None] :
_Maximum Time Warning [None] :
_Maximum Time Job [None] :
_LBAL Group [None] :
_Synchronization [None] :
_Comment [None] : This job extracts nightly notes
```

```
%NSCHED-I-RQSTSUCCSS, Job 1 - Created
```

You could also use **/NONOTIFY** with **SCHEDULE CREATE** or **SCHEDULE MODIFY**, as follows:

```
SCHEDULE> CREATE "@user$s:[smith]foo.com" -  
_SCHEDULE> /INTER="+0 00:30" -  
_SCHEDULE> /GROUP=personal -  
_SCHEDULE> /TYPE=daily -  
_SCHEDULE> /NAME=myjob  
_SCHEDULE> /NONOTIFY/NOPROMPT  
%NSCHED-I-RQSTSUCCSS, Job 2 - Created
```

How do I send mail notification to a distribution list instead of a single user? Use the following command:

```
SCHEDULE> MODIFY job_name  
/MAIL_ADDRESS="@device:[directory]filename.DIS"
```

The default location for files referenced in the manager (such as output files and files referenced in the OpenVMS command) is the owner's SYS\$LOGIN. You must specify the exact device and directory if the file is not in SYS\$LOGIN.

You must enclose the mail distribution list in quotation marks.

E.3.4. Stopped Jobs

What happens if some jobs are running in the manager when the manager fails or is stopped?

After the manager starts a job, the job is an independent process that continues to run even if you stop the manager with the **SCHEDULE STOP** command.

- If you restart the manager before the job completes, all is normal.
- If the job completes while the manager is stopped, the termination message with the accounting information is not written to the manager's mailbox, and the database is not updated. The job remains marked as running on the current node and process (PID).
- When the manager restarts, it reads through the database and verifies that all jobs marked as currently running are actually running. If the manager finds a job that is not running even though the database entry says it is, then the current time and date are filled in and the completion status is changed to:

```
NSCHED-F-ABORTED, Job was aborted
```

This is a failure status. Any mail or broadcast notification is sent at this point. If the **SET RESTART** command is enabled for that job, then the job runs again. Otherwise, the job is put on hold.

Whenever I stop and restart the manager, it creates a file named *your-system-name*_REMOTE.LOG. Is this normal?

Job Management Manager commands that come from outside your cluster are run by the SCHED_REMOTE process. This process receives DECnet connects, validates proxy accounts, and writes information to NSCHED\$:NODE_REMOTE.LOG to provide an audit trail of remote activity. When the SCHED_REMOTE process starts, an empty file is created.

You can edit the Job Management Manager startup procedure not to start the SCHED_REMOTE process if all of the following are true:

- You do not want to use Job Management Manager commands on remote nodes.
- The remote nodes do not need to communicate with you.
- You do not have any jobs on your cluster that depend on remote jobs.

E.3.5. Reports

What reports are available with the manager, and how do I invoke them?

The following reports are available:

- Future job runs (**SCHEDULE SHOW SCHEDULED_RUNS** command)
- Past job runs (**SCHEDULE SHOW HISTORY** command)
- Past activity in the manager (Run `NSCHED$:VSS_REPORTS.EXE.`)

How do I create a report that will show me what jobs are scheduled to run during the next eight hours?

Use the following command:

```
SCHEDULE> SHOW SCHEDULED_RUNS/END="+8"/FULL
```


Glossary

Glossary Topics

Cluster Alias	A collective name representing all nodes in one cluster.
Day Specifiers	The day(s) of the week that Job Management Manager runs the job.
Event Log	A file in which Job Management Manager events are logged, such as jobs starting, completing, resources consumed, and interface requests. The event log may be set to capture up to 11 event classes for each individual node in an OpenVMS Cluster. A log reader program provided with the kit produces standard reports.
Failover	When a Job Management Manager node crashes on an OpenVMS Cluster that has multiple nodes running Job Management Manager, one of the other nodes always updates and cleans up any jobs that were running on the crashed node and restarts the jobs if necessary.
Group	A job category, such as daily, weekly, or monthly. Users can specify jobs by group, and Job Management Manager can perform certain actions on the entire group at once.
Interval	See Schedule Interval.
Job	A job consists of several elements: an OpenVMS command to execute the job, information about how and when to run the job, and actions to take on job completion. The information includes interval, completion notification, output-file specification, and job dependencies. A job can also include a prefunction and postfunction.
Job Dependencies	Job(s) that must complete successfully before another job can run.
Job Management Manager Delta Time	The time interval from the current date and time to a date and time in the future. The Job Management Manager delta time format is: <i>+dddd hh:mm:ss.cc</i> .
Job Specifier	A job specifier reference a job through the DCL interface. A job specifier value is either a job name or a job number.
Load Balance Group	A Load Balance Group holds a subset of nodes in an OpenVMS cluster, for load balancing. A job associated to a Load Balance Group is restricted to run only on the nodes in that group and load balances among them.
Load Balancing	All jobs that are not restricted to run on one specific node in an OpenVMS Cluster normally run on the default node in the cluster. If load balancing is enabled, such unrestricted jobs can be assigned to run on other nodes in the OpenVMS Cluster. Each job is assigned to the least loaded node in the cluster or the least loaded node in the load balance group (if defined).
Management Commands	Management Commands require at least OPER or SYSPRV privileges to use. The Job Management Manager management commands

perform the following tasks: stop the Job Management Manager from running on the entire OpenVMS Cluster, on a specific node in the cluster, or on a specific remote node on the network; set and show certain Job Management Manager features; delete pending job-dependency notification retries on remote nodes; and close, rename, display, or print event logs.

Next Scheduled Time	A Job Management Manager job's next scheduled run time.
OPCOM	Operator Communication Manager: A system process that controls communications between user processes and a system operator.
OpenVMS Delta Time	The time interval from the current date and time to a date and time in the future. OpenVMS delta time format is: <i>dddd-hh:mm:ss.cc</i> .
Postfunction	An OpenVMS command that is executed after a job's main OpenVMS command. Each job may have its own postfunction. A postfunction runs even if the main command fails.
Prefunction	An OpenVMS command that is executed before a job's main OpenVMS command. Each job may have its own prefunction. If the prefunction ends with an error, the main command does not run, and the entire job terminates with an error.
Proxy Account	An account, defined in the Network User Authorization File that allows a remote user to have access to the local node by mapping into a local account. The remote user assumes all privileges and rights of the local account. Job Job Management Manager remote access is allowed through both default and specified proxies or through an explicit account and password specification.
Remote Node	Any node in the network that is not in the user's local OpenVMS Cluster.
Remote Node Job	A job designated to run on a remote Job Management Manager agent that is running on a remote machine.
Schedule Interval	The time period between a job's start times. The time period can be daily, hourly, Job Management Manager delta time, fiscal time, regular monthly calendar interval, and continuous run.
Special Days	Jobs in a Special Day Class have their run times restricted by the class. Such jobs can be set to run only on days in the class, or can be set not to run on such days and be skipped or rescheduled for the next available day. Only users with OPER or SYSPRV privileges can create, modify, or delete Special Day Classes.
Type	A job category, such as backup, fiscal, or documentation. You can specify jobs by type, and the Job Management Manager can perform actions on jobs by type.
Wildcard Characters0	The asterisk (*) and percent (%) characters are wildcards. The asterisk (*) matches any number of characters. The percent (%) matches exactly one character.