

JAMS Reference Manual

Order Number: JAMS-RM-42

This manual provides a complete description of the *Job Access & Management System*.

Revision/Update Information:	This manual supersedes the <i>Guide to the Job Access & Management System, Version 4.0</i> .
Operating System and Version:	OpenVMS/VAX Version 6.2 or higher, OpenVMS/AXP V6.2 or higher
Software Version:	JAMS Version 4.2

The information in this document is subject to change without notice and should not be construed as a commitment by MVP Systems Incorporated. MVP Systems, Inc. assumes no responsibility for any errors that may appear in this document.

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

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

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

If you have questions about JAMS please feel free to call JAMS technical support at (866) 259-5267. Technical support is available around the clock, 24 hours a day, 7 days a week. The mailing address is:

TECHNICAL SUPPORT DEPT.
MVP SYSTEMS INC.
2700 E. MAIN ST.
SUITE 108
COLUMBUS, OH 43209

E-Mail: JAMS@mvpsi.com
WWW: <http://JAMS.mvpsi.com>
WWW: <http://www.mvpsi.com>

The following are trademarks of Hewlett-Packard Company: Alpha, DECnet, DECwindows, OpenVMS, VAX, VMS, VMScluster.

Window and Windows NT are registered trademarks of Microsoft Corporation.

Motif is a registered trademark of The Open Group, Inc.

Contents

PREFACE

x

CHAPTER 1 THE JAMS DATABASE

1-1

1.1 THE JAMS DATABASE

1-1

1.2 SYSTEM DEFINITIONS

1-2

1.2.1	System I.D.	1-2
1.2.2	Description	1-2
1.2.3	Security	1-2
1.2.4	Notify Names	1-3
1.2.5	Notify Options	1-5
1.2.6	Submit Options	1-6
1.2.7	LOG File Options	1-7
1.2.8	Resources	1-7
1.2.9	Print Options	1-7
1.2.10	Source Options	1-8
1.2.11	Documentation Location	1-9
1.2.12	Recovery Instructions	1-9

1.3 JOB DEFINITIONS

1-10

1.3.1	Job Name	1-10
1.3.2	Description	1-10
1.3.3	System	1-10
1.3.4	Security	1-10
1.3.5	Submit Options	1-11
1.3.6	Submit Times	1-12
1.3.7	Schedule Options	1-13
1.3.8	Parameters	1-14
1.3.9	Reports	1-16
1.3.10	Dependencies	1-18
1.3.10.1	Job Completion Fields •	1-18
1.3.10.2	Remote Job Completion Fields •	1-19
1.3.10.3	Variable Value Fields •	1-21
1.3.11	Source Options	1-21
1.3.12	View Source	1-22
1.3.13	Edit Source	1-22

Contents

1.3.14	Precheck Job	1-22
1.3.15	Recovery Job	1-23
1.3.16	Recovery Instructions	1-24
1.3.17	Notify Names	1-24
1.3.18	Notify Options	1-26
1.3.19	Log Options	1-28
1.3.20	Resources	1-28
1.3.21	Execution Statistics	1-28
1.3.22	Completion Dates	1-29
1.3.23	References	1-29
1.4	SETUP DEFINITIONS	1-30
1.4.1	Setup Name	1-30
1.4.2	Description	1-30
1.4.3	System	1-30
1.4.4	Security	1-30
1.4.5	Submit Options	1-31
1.4.6	Submit Times	1-32
1.4.7	Schedule Options	1-33
1.4.8	Jobs	1-34
1.4.9	Resources	1-36
1.4.10	Dependencies	1-37
1.4.11	Precheck Job	1-37
1.4.12	Recovery Job	1-38
1.4.13	Recovery Instructions	1-39
1.4.14	Notify Names	1-39
1.4.15	Notify Options	1-41
1.4.16	Log Options	1-41
1.4.17	Execution Statistics	1-42
1.4.18	Completion Dates	1-42
1.5	TRIGGER DEFINITIONS	1-43
1.5.1	Events	1-44
1.5.1.1	Job Completion Events •	1-44
1.5.1.2	Variable Value Event Fields •	1-44
1.5.2	Actions	1-44
1.6	VARIABLE DEFINITIONS	1-46
1.7	MENU DEFINITIONS	1-48
1.7.1	Menu Detail	1-48

1.8	NAMED TIMES	1-50
1.9	DATE TYPES	1-51
1.10	DATE DEFINITIONS	1-53
1.11	CONFIGURATION	1-55
1.12	NODE DEFINITIONS	1-59
1.13	NODE GROUPS	1-60
1.14	ACCESS CONTROL	1-61
1.14.1	History Inquiry	1-61
1.14.2	Job Monitor Access	1-61
1.14.3	Setup Definitions	1-62
1.14.4	Job Definitions	1-62
1.14.5	System Definitions	1-63
1.14.6	Menu Definitions	1-63
1.14.7	Variable Definitions	1-64
1.14.8	Trigger Definitions	1-64
1.14.9	Date Definitions	1-64
1.14.10	Configuration	1-64
1.14.11	Date Type Definitions	1-65
1.14.12	Access Control	1-65
1.14.13	Node Definitions	1-65
1.14.14	Named Times Access	1-65
1.14.15	Username Access	1-66
1.14.16	Resource Access	1-66
CHAPTER 2 JAMS COMMANDS		2-1
2.1	COMMAND ENVIRONMENT	2-1
	ACQUIRE RESOURCE	2-2
	CREATE DATE	2-3
	CREATE DTYPE	2-5
	CREATE METHOD	2-6
	CREATE RESOURCE	2-8
	CREATE TIME	2-10
	CREATE VARIABLE	2-12

Contents

DEFINE JOB	2-14
DEFINE MENU	2-18
DEFINE SETUP	2-19
DEFINE SYSTEM	2-24
DEFINE TRIGGER	2-26
DELETE ENTRY	2-28
DELETE JOB	2-29
DELETE MENU	2-30
DELETE METHOD	2-31
DELETE RESOURCE	2-32
DELETE SETUP	2-33
DELETE SYSTEM	2-34
DELETE TIME	2-35
DELETE TRIGGER	2-36
DELETE VARIABLE	2-37
DISABLE TIME	2-38
DISABLE TRIGGER	2-39
ENABLE TIME	2-40
ENABLE TRIGGER	2-41
EVALUATE DATE	2-42
EXIT	2-43
EXTRACT DATE	2-44
EXTRACT DTYPE	2-45
EXTRACT JOB	2-46
EXTRACT MENU	2-47
EXTRACT METHOD	2-48
EXTRACT RESOURCE	2-49
EXTRACT SECURITY	2-50
EXTRACT SETUP	2-52
EXTRACT SYSTEM	2-53
EXTRACT TIME	2-54
EXTRACT TRIGGER	2-55
EXTRACT USERNAME	2-56
EXTRACT VARIABLE	2-57
FAKE COMPLETION	2-58
GET STATUS	2-59
GET VARIABLE	2-60
HELP	2-61
MENU	2-62
RECORD	2-65
RELEASE RESOURCE	2-66
RENEW LOG_FILE	2-67
RESET JOB STATISTICS	2-68
RESET SETUP STATISTICS	2-69
RESET TRIGGER	2-70
SET AGENT	2-71
SET ENTRY	2-72
SET LICENSE	2-74
SET RESOURCE	2-75
SET SECURITY	2-76
SET STATUS	2-77
SET USERNAME	2-78
SET VARIABLE	2-80

SHOW ENTRY	2-82
SHOW LICENSE	2-84
SHOW RESOURCE	2-85
SHOW STATUS	2-86
SHOW TIME	2-87
SHOW TRIGGER	2-88
SHOW VARIABLE	2-89
START MONITOR	2-90
START NETWORK	2-91
START SCHEDULE	2-92
STOP MONITOR	2-93
STOP NETWORK	2-94
STOP SCHEDULE	2-95
SUBMIT	2-96

CHAPTER 3 JAMS JOBS 3-1

3.1 REPORT HEADERS 3-1

3.2 JOBS SUPPLIED WITH JAMS 3-1

JAMS_AUTOSUBMIT	3-2
JAMS_AUTOSUBMIT_INTER.EXE	3-4
JAMS_CLEANUP	3-6
JAMS_PURGE	3-7
JAMS1000	3-8
JAMS1001	3-9
JAMS1100	3-10
JAMS1101	3-11
JAMS1200	3-13
JAMS2000	3-14
JAMS2100	3-16
JAMS3000	3-17

CHAPTER 4 USING THE JAMS CALLABLE INTERFACE 4-1

4.1 USING THE CALLABLE INTERFACE 4-1

4.1.1	Calling JAMS Routines	4-1
4.1.2	Using Item Lists	4-1
4.1.3	Linking with JAMSSHR.EXE	4-2
4.1.4	Privileges	4-2

4.2 SAMPLE PROGRAMS 4-3

4.3	ROUTINES	4-3
	JAMS_EVALUATE_DATE	4-4
	JAMS_GET_DATA	4-6
	JAMS_GET_JOB_STATUS	4-22
	JAMS_HISTORY_FREE	4-23
	JAMS_HISTORY_GET	4-24
	JAMS_HISTORY_INIT	4-25
	JAMS_GET_VARIABLE	4-28
	JAMS_SCHCALLUSER	4-29
	JAMS_SCHEDULE_FREE	4-31
	JAMS_SCHEDULE_GET_DEPEND	4-32
	JAMS_SCHEDULE_GET_JOB	4-33
	JAMS_SCHEDULE_INIT	4-34
	JAMS_SCHEDULE_SORT	4-37
	JAMS_SET_DATA	4-38
	JAMS_SET_JOB_STATUS	4-40
	JAMS_SET_VARIABLE	4-41
	JAMS_SUBMIT_MENU	4-43
	JAMS_SUBMIT_JOB	4-45

CHAPTER 5 TEMPLATES AND JOB PARSING 5-1

5.1	COMMAND FILES	5-1
5.1.1	Command File Template Expansion	5-1
5.2	HOW TEMPLATES ARE EXPANDED	5-3
5.2.1	Simple Parameter References	5-4
5.2.2	Qualified Parameter References	5-6
5.2.3	Function References	5-7
5.2.4	Controlling Report Printing	5-9
	5.2.4.1 Base Functionality • 5-9	
	5.2.4.2 Printing a Report More than Once • 5-9	
	5.2.4.3 Controlling Print Time • 5-9	
5.2.5	An Example of Template Expansions	5-9
5.2.6	Trapping DCL errors	5-13

APPENDIX A ERROR MESSAGES IN ORDER BY IDENTIFIER A-1

A.1	ERROR MESSAGE FORMAT	A-1
------------	-----------------------------	------------

APPENDIX B ERROR MESSAGES IN ORDER BY TEXT
B-1

B.1
B-1

ERROR MESSAGE FORMAT

INDEX

EXAMPLES

3-1	Report Header	3-1
3-2	JAMS_AUTOSUBMIT Report	3-3
3-3	JAMS1000 Report	3-8
3-4	JAMS1001 Report	3-9
3-5	JAMS1100 Report	3-10
3-6	JAMS1101 Report	3-12
3-7	JAMS1200 Report	3-13
3-8	JAMS2000 Report	3-15
3-9	JAMS3000 Report	3-18
5-1	JAMS_MAIN Template Module	5-4
5-2	Sample Jobs command file	5-11
5-3	Sample Job's command file after parsing	5-11
5-4	Trapping DCL errors	5-13

TABLES

5-1	JAMS Predefined Parameters	5-5
5-2	Data Types and Formats	5-6
5-3	Data Classes	5-7
5-4	Data Class Parameters	5-8

Preface

This manual provides a detailed description of the capabilities of the *Job Access & Management System (JAMS)*. The information contained within this manual is of interest to those who are responsible for installing, implementing and using this software package.

Additional Documentation

JAMS is designed to run under the OpenVMS operating system and this manual assumes that the reader has a basic understanding of OpenVMS. The following OpenVMS documentation may also be helpful when using *JAMS*.

- *Guide to using Command Procedures* provides information on command procedures and batch jobs.
- *DCL Dictionary* provides information on the SUBMIT and PRINT commands and their qualifiers.
- *Guide to Programming Resources* provides information on developing programs in the OpenVMS environment.
- *Guide to OpenVMS System Security* provides information on Access Control List (ACL) based security.

Conventions

The following conventions are used in this manual:

Convention	Description
UPPERCASE	Uppercase words and letters used in examples indicate text that you should type exactly as shown.
lowercase	Lowercase words and letters used in examples indicate text that you should substitute a word or value of your choice.
[]	Brackets ([]) indicate optional elements.
{ }	Braces ({}) indicate required elements.
...	An ellipse (...) indicates that the preceding item can be repeated.
CTRL/x	The text CTRL/x indicates that you should press and hold the key labeled Ctrl while you press the key indicated by the x (for example CTRL/C, CTRL/Z).
Key Name	Text enclosed in a box indicates that you should press the key whose name is enclosed by the box.
Gold/x	Key names which begin with Gold/ indicate that you should first press and release the Gold key (PF1 on most keyboards) and then press the key indicated by the x.

1 The JAMS Database

This chapter describes the *JAMS* Database.

1.1 The JAMS Database

Without entering any information into the *JAMS* database, you can immediately use *JAMS* to monitor, manage, and record job execution history for all OpenVMS based batch jobs.

By making just a few entries in the *JAMS* configuration, you can turn on *JAMS* notifications which will immediately alert support personnel when jobs terminate abnormally.

In order to fully utilize *JAMS* on OpenVMS, Windows NT, and Unix/Linux you must create Job definitions in *JAMS*. These Job definitions can be created by using the *JAMS* Screen, Motif, or Command line interfaces.

1.2 System Definitions

Systems are used to place Jobs into logical groups. The items specified in a System definition apply to all of the Jobs in the System. System security is very important. System security defines who may manipulate Job definitions within the System and who may submit Jobs for processing.

Systems can be maintained with the screen, command or Motif based interfaces.

1.2.1 System I.D.

The System I.D. is a unique identifier for a System. A System I.D. must be a valid OpenVMS identifier. This means that it must begin with an alphabetic character and contain only the characters A-Z, 0-9, \$ (dollar sign), and _ (underscore). Also the last character may not be an underscore.

1.2.2 Description

The description is used in menus, lists and reports to provide a more complete description of the System than provided by the System I.D.

1.2.3 Security

If you select the *Security* option and you have CONTROL access to the System (or the OpenVMS BYPASS privilege), the System's Access Control List will be displayed. You can then make modifications to the ACL. The following access rights can be specified:

Right	Meaning
CONTROL	Allows modification of this System's Access Control List.
MONITOR_JOBS	Allows Jobs in this System to display in the Job Monitor.
CHANGE	Allows modification of this System definition provided that the user also has CHANGE access to System Definitions.
INQUIRE	Allows inquiry into this System definition provided that the user also has INQUIRE access to System Definitions.
DELETE	Allows deletion of this System definition provided that the user also has DELETE access to System Definitions.
SUBMIT	Allows submission of Jobs in this System.
DEBUG	Allows submission of Jobs in this System but only if the /DEBUG qualifier is used on the JAMS SUBMIT command. This qualifier will submit the Job under the users OpenVMS username rather than the OpenVMS username specified in the System Definition.
JOB_ADD	Allows addition of new Job Definitions to this System.
JOB_CHANGE	Allows modification of existing Job definitions in this System.
JOB_INQUIRE	Allows inquiry into existing Job definitions in this System.
JOB_DELETE	Allows deletion of existing Job definitions in this System.
DEFINE_SETUP	Allows manipulation of Setup Definitions for Jobs in this System.
OPERATOR	Allows the user to use the JAMS Job Monitor to hold, reschedule, release or delete an occurrence of a Job in this system.
ABORT_JOBS	Allows the user to use the JAMS Job Monitor to abort or restart an occurrence of a Job in this system.

Note that in order to change a Job definition's System I.D., you must have JOB_DELETE access to the old System I.D. and JOB_ADD access to the new System I.D.

1.2.4 Notify Names

The *Notify Names* screen is used to define who should be notified when a Job in this System requires notification. Notification is required when:

- A Job completes with a severity which is worse than the Job's notification severity.
- A Job runs too long.
- A Job is stalled.
- A Job misses it's execution window.
- A Job runs too quickly.

The fields on this category include:

The JAMS Database

System Definitions

OpenVMS Mail Address

This is a list of OpenVMS Mail addresses, separated by commas. If a Job in this System requires notification, a OpenVMS mail message will be sent to this list of addresses.

OpenVMS REPLY Username List

This is a list of OpenVMS usernames, separated by commas. If a Job in this System requires notification, a message will be broadcast to these users (if they are logged on at the time.)

Operator Classes

This is a list of OpenVMS operator classes, separated by commas. If a Job in this System requires notification, a message will be sent via OPCOM to these operator classes.

Valid OpenVMS operator classes are:

CARDS
CENTRAL
CLUSTER
DEVICES
DISKS
NETWORK
OPER1 through OPER12
PRINTER
SECURITY
TAPES

Notification Job Name

This is the name of a *JAMS* Job which should be submitted when notification is required.

If the Notification Job is parsed then the following predefined parameters are available:

JAMS_NOTIFY_ENTRY	The OpenVMS queue entry number of the job which caused the notification.
JAMS_NOTIFY_JAMS_ENTRY	The <i>JAMS</i> entry number of the job which caused the notification.
JAMS_NOTIFY_JOB_NAME	The Job Name of the job which caused the notification.
JAMS_NOTIFY_JOB_STATUS	The current Job Status value of the job which caused the notification.
JAMS_NOTIFY_LOG_FILENAME	The full file specification of the log file for the job which caused the notification.
JAMS_NOTIFY_MAIL_ADR	The list of names which will be notified via OpenVMS Mail.
JAMS_NOTIFY_OPER_CLASSES	The list of OPCOM classes will receive an operator message.

JAMS_NOTIFY_PID	The process ID of the job which caused the notification.
JAMS_NOTIFY_REASON	The reason for the notifications. This parameter could be: COMPLETED, FAILED, SHORT, RUNAWAY, STALLED or MISSED_WINDOW.
JAMS_NOTIFY_REPLY_USERS	The list of names which will receive a broadcast message.
JAMS_NOTIFY_RUN	The JAMS run occurrence number of the job which caused the notification.
JAMS_NOTIFY_STATUS	The final status of the job which caused the notification.
JAMS_NOTIFY_SUBMITTED_BY	The username which submitted the job which caused the notification.
JAMS_NOTIFY_SYSTEM_ID	The JAMS System I.D. of the job which caused the notification.

1.2.5 Notify Options

Stalled Job

The Stalled Time specifies how much time may elapse after a jobs scheduled time before the job is considered to be stalled. When the job is considered stalled, *JAMS* will perform notification for the job.

Runaway Job Elapsed Time

The Runaway Job Elapsed Time Percent specifies how long a job may run before it is considered a runaway job. This is specified as a percentage of the jobs average elapsed time. When the job is considered runaway, *JAMS* will perform notification for the job.

Runaway Job CPU Time

The Runaway Job CPU Time Percent specifies how much CPU time a job may consume before it is considered a runaway job. This is specified as a percentage of the jobs average CPU time. When the job is considered runaway, *JAMS* will perform notification for the job.

Short Job Elapsed Time

The Short Job Elapsed Time Percent specifies a minimum elapsed time for a job. If the job completes successfully in less than this amount of time it is considered a *Short Job*. This is specified as a percentage of the jobs average elapsed time. When a job is determined to be a short job, *JAMS* will perform notification for the job and may change the completion severity of the job.

The JAMS Database

System Definitions

Short Job CPU Time

The Short Job CPU Time Percent specifies a minimum CPU time for a job. If the job completes successfully but used less than this amount of CPU time it is considered a *Short Job*. This is specified as a percentage of the jobs average CPU time. When a job is determined to be a short job, JAMS will perform notification for the job and may change the completion severity of the job.

Short Job Completion Severity

If specified, when a job is determined to be a *Short Job*, the completion severity will be changed to the severity specified here.

1.2.6 Submit Options

The *Submit Options* form is used to specify values which are used when Jobs in this System are submitted. The fields on this screen include:

Default Batch Queue

The default batch queue specifies to which OpenVMS batch queue Jobs in this System should be submitted. This queue can be overridden in the Job definition. If left blank, SYS\$BATCH is the default.

Retain in Queue

Specifies the default retention policy for Jobs in this System. The options are:

- Always - Jobs are always retained until specifically deleted.
- Never - Jobs are never retained.
- Error - Jobs are retained if they complete with a severity of Warning or worse.
- Timed - Jobs are retained for the specified time after completion.

Scheduling Priority Modifier

This number is used to calculate the scheduling priority for Jobs in this System. This number may range from -127 to 127, the default being 0. When a Job is submitted by the JAMS submit sub-system, the scheduling priority is calculated by adding together the OpenVMS default scheduling priority, the System's priority modifier and the Job's priority modifier.

Submit under OpenVMS Username

This field specifies a valid OpenVMS username under which Jobs in this System will be submitted. You can use this feature to limit a persons access capabilities to only the capabilities of the Jobs within a System. For example, you could grant a user the ability to submit jobs in the PAYROLL System, but use OpenVMS file protection to prevent them from accessing any of the Payroll files. When the user submits a Payroll Job, the Job would execute under a different OpenVMS username which would have access to the Payroll files.

You must have either the CMKRNL privilege or SUBMIT access to the Username which you specify. See the JAMS SET USERNAME command for information on setting access to Usernames.

1.2.7 LOG File Options

The *Log File Options* screen is used to specify how .LOG files should be handled. These options apply to all Jobs in this System. The fields on this screen include:

Keep Logs

When the field is true, or “Y”, batch log files will not be deleted at the completion of the Job.

Print Logs

When the field is true, or “Y”, batch log files will be printed when the Job completes.

Time Stamp .LOG File Names

When the field is true, or “Y”, batch log files be time stamped with an extension of .LOG_yyyymmdd_hhmmsscc.

Log Location

This field specifies the OpenVMS device and directory where batch log files should be placed. If left blank, they are placed in SYS\$LOGIN.

Note: If you specify a Log File Directory, you must make sure that the OpenVMS username which the Jobs in this System will execute under has write access to the directory. Also, if the files in the specified directory have a version limit, the OpenVMS username must have delete access to the oldest version of a log file (assuming that the version limit has been reached).

Print Queue

This field specifies which OpenVMS print queue batch log files should be printed on, if they are printed at all. If left blank, SYS\$PRINT is used.

1.2.8 Resources

The *Resources* option will display a list of the Resource requirements which are currently defined for this System. All of the Jobs and Setups in this System will inherit these Resource requirements.

Resource requirements can also be specified at the Job or Setup level.

1.2.9 Print Options

The *Print Options* option is used to set values which are used when a Job has defined Reports. The fields on this screen include:

The JAMS Database

System Definitions

Default Print Queue

The default print queue specifies to which OpenVMS print queue Reports in this System should be printed. Report definitions are part of a Job's definition. This queue can be overridden in the Report definition. If left blank, SYS\$PRINT is used.

Location for Print Files

This field specifies the OpenVMS device and directory where print files should be placed. If left blank, they are placed in SYS\$LOGIN. This field is only used by Jobs which are parsed.

1.2.10 Source Options

The *Source Options* option is used to set values which govern where job source files are located and how they are accessed. The fields on this screen include:

Location of Job Source Files

This field specifies the OpenVMS device and directory where Job Source files are located. This may also be a logical name which equates to a device and directory.

Is Location Required

If the location is required then the device/directory specified in the "Job Source Location" field is the required location. A device/directory specification in a Job Definition will be ignored.

If the location is not required, then the value in the "Job Source Location" field is used as a default which may be overridden in a Job Definition.

Use Secure Logical Names

A secure logical name is one which is defined with the /EXECUTIVE qualifier.

If Jobs in this System will run under a privileged username, we strongly recommend that you use only secure logical names.

Template Library

This field specifies the full filename of a OpenVMS text library which contains the JAMS templates used when parsing a Job. Refer to Chapter 5, Templates and Job Parsing for a complete description of template libraries. If you do not specify a complete filename, the unspecified portions are taken from the file specification "JAMS_DATA:JAMS_TEMPLATE.TLB"

Any logical names referenced by this specification must be defined with the /EXECUTIVE qualifier.

1.2.11 Documentation Location

The *Documentation Location* option is used to specify the device/directory where documentation files for this System should be located.

With this version of *JAMS*, documentation files are limited to restart/recovery instructions for Systems and Jobs.

1.2.12 Recovery Instructions

The *Recovery Instructions* option is used to edit or view this System's restart/recovery instructions. These instructions are included in the OpenVMS mail message which is sent if a Job in this System fails.

You can also create restart/recovery instructions at the Job level.

1.3 Job Definitions

Job Definitions define a single command file's attributes. The Job definition serves to document the command file's existence as well as providing the capability of submitting the command file using the *JAMS* Submit sub-system.

1.3.1 Job Name

The Job Name is a unique identifier for a Job. It is analogous to the filename of a command file.

A Job Name must be a valid OpenVMS identifier. This means that it must begin with an alphabetic character and contain only the characters A-Z, 0-9, \$ (dollar sign), and _ (underscore). Also the last character may not be an underscore.

1.3.2 Description

The *Description* option is used to view or update the Job's description. The description is used in menus, lists and reports to provide a more complete description of the Job than provided by the Job Name. Two lines of 50 characters are provided. If you use only one line, use the first one.

1.3.3 System

The *System* option is used to view or update the Job's System I.D. The System I.D. specifies to which System this Job definition belongs. This is a required field and must be a System which has already been defined using the System Definitions option.

Some of the information from the selected System Definition is also displayed on this form.

1.3.4 Security

If you select the *Security* option and you have CONTROL access to the Job (or the OpenVMS BYPASS privilege), the Job's Access Control List will be displayed. You can then make modifications to the ACL. The following access rights can be specified:

Right	Meaning
CONTROL	Allows modification of this Job's Access Control List.
CHANGE	Allows modification of this Job definition provided that the user also has CHANGE access to Job Definitions.
INQUIRE	Allows inquiry into this Job definition provided that the user also has INQUIRE access to Job Definitions.
DELETE	Allows deletion of this Job definition provided that the user also has DELETE access to Job Definitions.
SUBMIT	Allows submission of this Job.
DEBUG	Allows submission of this Job but only if the /DEBUG qualifier is used on the JAMS SUBMIT command. This qualifier will submit the Job under the users OpenVMS username rather than the OpenVMS username specified in the System Definition.
OPERATOR	Allows the user to use the JAMS Job Monitor to hold, reschedule, release or delete an occurrence of this Job.
ABORT_JOBS	Allows the user to use the JAMS Job Monitor to abort or restart an occurrence of this Job.

Note that in order to change a Job definition's System I.D., you must have JOB_DELETE access to the old System I.D. and JOB_ADD access to the new System I.D.

1.3.5 Submit Options

The *Submit Options* option is used to view or update fields associated with submitting this Job. The fields on this screen include:

Override Batch Queue

This field is used to specify the name of the batch queue where this Job should be submitted. This entry will only be used if the execution method for the Job utilizes batch queues. If not specified, the Default Batch Queue from the System Definition is used.

Scheduling Priority Modifier

This number is used to calculate the scheduling priority for this Job. This number may range from -127 to 127, the default being 0. When a Job is submitted by the JAMS submit sub-system, the scheduling priority is calculated by adding together the OpenVMS default scheduling priority, the System's priority modifier and the Job's priority modifier.

Retain in Queue

Specifies the retention policy for this Job. The options are:

Always	Job is always retained until specifically deleted.
Never	Job is never retained in the queue.

The JAMS Database

Job Definitions

Error	Job is retained if they complete with a severity of Warning or worse.
Timed	Job is retained for the specified time after completion.
Not Specified	Use the default specified in the System Definition.

Hold job when submitted

This field specifies whether or not this Job should be placed in a “holding” state when it is submitted.

This is useful for Jobs which are requested, or submitted, by end-users but require operator approval before execution.

Restartable

This field specifies whether or not the Job can be restarted after a failure.

Suppress Menu Display

This field specifies whether or not this Job should be excluded from *JAMS* Submit menus. If set to “Yes”, the job will never be displayed on a Submit menu. The only way to submit the Job is via the *JAMS SUBMIT* command or with a Setup definition.

Notify Submitter

This field specifies whether or not the /NOTIFY qualifier is used when submitting this Job. Note that when /NOTIFY is specified, the username who submitted the job is notified, not the username under which the job ran.

Run Under Username

You can specify the username under which this Job should run. This will override the Username specified in the Job’s System Definition. You must have either the CMKRNL privilege or SUBMIT access to the Username which you specify. See the *JAMS SET USERNAME* command for information on setting access to Usernames.

Agent Node Name

For Jobs which execute on remote nodes, this is the name of the system where they will execute. This system must be running either the *JAMS* Agent or an rexec server (depending on the Execution Method used by the Job).

1.3.6 Submit Times

Submit Window Time

This field is used to specify the name of a *JAMS* Named Time which defines this Job’s Submit Window. You can specify a Named Time or a specific time range but not both.

Job may be scheduled between

These fields are used to specify a specific Submit Window for this Job. You can specify a Named Time or a specific time range but not both.

Missed Window Action

This field specifies the action that *JAMS* should take if this Job misses it's time window. Possible actions are:

No Action	The window is only used to limit the time which can be entered when the Job is manually submitted.
Continue	The Job will not start until the time window opens but will not be rescheduled or aborted if the window closes before the Job completes.
Abort or Delete	The Job will not start until the time window opens and will be deleted or aborted if the window closes before the Job completes.
Reschedule or Continue	The Job will not start until the time window opens. If the Job has not started when the window closes, it will be rescheduled for the next time that the window opens. If it has started, it will be allowed to continue.
Restart or Reschedule	The Job will not start until the time window opens. If the Job has not complete when the window closes, it will be rescheduled for the next time the window opens. If the Job is executing when the window closes it will be aborted and rescheduled.

Notify of Missed Window

Defines whether or not *JAMS* should perform notification if the Job misses it's time window.

Default Submit Date

This field is used to specify a default submit date which is used when this Job is manually submitted.

Default Submit Time

This field is used to specify a default submit time which is used when this Job is manually submitted.

1.3.7 Schedule Options

The *Schedule Options* option is used to define a scheduled time for this Job. You can also use a Setup Definition to schedule a Job.

Scheduled Date

This is an English language representation of the date, or dates, on which this Job is scheduled to run.

You can specify multiple scheduled days by separating the text with commas. For example, "MONDAY,WED,FRIDAY" means that this Job should be run every Monday, Wednesday and Friday.

The JAMS Database

Job Definitions

Note that the quotations shown in the examples should not be entered in the scheduled date field. For a complete description of English language date representations, refer to Chapter 7 of the JAMS User Guide.

Except for

This is an English language representation of a date, or dates, on which this Job should *not* be run. If you want to schedule a Job to run every Monday, except for the last Monday of the month, you would enter “MONDAY” into the *Scheduled Date* field and “LAST MONDAY OF MONTH” into this field.

Scheduled Time

This is the time of day when this Job should be run.

Every hh:mm Until

If you want this Job to run many times during a day, you can specify the delta time between runs and the time of day to stop repeating. Then, any time this Job completes it will be resubmitted to run *hh:mm* later. The Job will be resubmitted no matter what its completion status was or how it was submitted. The only way to stop this repetition is to abort/delete the scheduled entry.

You can specify what time to use as the base for calculating the next run time, the base may be the Scheduled, Start or End time.

Automatic Submit at Scheduled Date and Time

This field indicates whether or not this Job should be automatically submitted at the scheduled date and time.

A no entry indicates that the scheduled date and time are to be used only for management and capacity planning.

Non-workday Scheduling

This field indicates how this Job should be scheduled if the day it is scheduled to run falls on a non-workday. There are three options:

- S - Schedule the Job, even on non-workdays.
- I - Ignore the Job, do NOT schedule on non-workdays.
- D - Defer the Job until the next workday.

1.3.8 Parameters

A Job may have zero or more parameters. When the JAMS Submit sub-system is used to submit a job which has parameters, the person submitting the job is presented with a “fill in the blanks” form which they use to provide values for the Job’s parameters.

Normal OpenVMS batch Jobs are limited to eight parameters but a JAMS parsed Job may have up to 255 parameters. If a job is not parsed, parameters other than P1 through P8 are of very little value. If a job is parsed, you can embed parameter names into the job’s DCL command file and JAMS will replace the embedded parameter name with the value of

the parameter. Please refer to Chapter 5, Templates and Job Parsing for a complete description of Job parsing, template expansion and parameters.

Note that the order of parameters is important. When *JAMS* builds a parameter form, the parameters are placed on the form in the order in which they are defined.

Parameter Name

This is the name for the parameter. It must be unique within the Job.

For OpenVMS jobs, the names P1, P2, P3, P4, P5, P6, P7 and P8 have special significance. They are treated exactly the same as any other parameter, but they are also specified as parameters to the Job as though you had used the /PARAMETER qualifier of the OpenVMS SUBMIT command.

Parameter Names must be valid symbol names. They must begin with a alphabetic character and contain only the characters A through Z, 0 through 9, \$ (dollar sign) and _ (underscore). In addition, a symbol name may *not* begin with the character string "JAMS_". Symbols which begin with this string are reserved for *JAMS*.

Prompt

When a job is submitted by *JAMS*, a form is created which is used to prompt the user for the values of the job's parameters. This text will be right justified and used as a prompt on this parameter form.

Data Type

This field specifies the data type of the parameter. Valid entries are:

- TEXT - A simple text field.
- INTEGER - An integer (entered in base 10).
- DATE - A valid date.
- TIME - A valid time of day.

Length

If the field has a data type of TEXT or INTEGER, you must specify the maximum length of the field.

Default Output Format

This field is the default edit string to be used when formatting this parameter during Job parsing. This field does not affect what the end-user sees or enters. It only affects how the parameter is passed to the Job. This field can be overridden in a parsed job.

This field applies only to fields with a data type of DATE, TIME or INTEGER.

Required

This is a Y (yes) or N (no) field which specifies whether or not an entry into this field is required.

The JAMS Database

Job Definitions

Mustfill

This is a Y (yes) or N (no) field which specifies whether or not the user must completely fill this field.

Uppercase

This is a Y (yes) or N (no) field which specifies whether or not the data entered is converted to uppercase upon entry.

Allow Entry

This is a Y (yes) or N (no) field which specifies whether or not the the end-user will be allowed to make an entry into this parameter when they are submitting the Job.

Hide

This is a Y (yes) or N (no) field which specifies whether or not the this parameter will be displayed when the end-user is submitting this Job.

Default Variable Name

This field specifies the name of a *JAMS* Variable whose value will be used as the default value for this parameter when the Job is submitted. If you enter a Variable Name here, you will not be able to enter a value in the Default field.

Default

This field specifies the default value for this field.

The default value for date fields may be entered as a specific date or using *JAMS* English language date text such as "LAST MONDAY", "FIRST MONDAY OF MONTH" etc. For more information on this syntax, refer to Chapter 7 of the *JAMS* User Guide.

Help

The text entered here is displayed when the user presses the **Help** key while the cursor is located in this parameters field.

1.3.9 Reports

Report definitions are used to document and control the reports produced by a Job. For Jobs which are not parsed by *JAMS*, Report definitions serve as documentation only. Jobs which are parsed, make full use of Report definitions. During the parsing, a logical name is defined which translates into a full file specification for the Report's print file. OpenVMS PRINT commands are also generated which send these print files to a printer queue. Refer to Chapter 5, Templates and Job Parsing for a complete description of Job parsing and template expansion.

A Job may have one or more Report definitions. There is no practical limit to the number of report definitions which a Job may have.

Report I.D.

The Report I.D. is an identifier used to name a report. It must be unique within a Job.

Description

This field provides a description of the Report which is used in the Report list, and can be used in the heading of the actual printed report.

Logical Name

When *JAMS* parses a Job, it can create the DCL command needed to define logical names which point to the actual report listing file. This field specifies this Report's logical name. If you leave this field blank, the Report I.D. is used as the value for the logical name.

This is also used when a Job needs to print the same report more than one time, such as to two different print queues. To accomplish this, define two Reports for the Job (which must have different Report I.D.s) but enter the same value in the logical name field for both Reports. When *JAMS* parses the .JOB file, it will define the logical name only once but it will generate two print commands, one for each Report.

Print Queue

This field specifies the OpenVMS print queue where the Report should be printed at the conclusion of the Job. If you leave this field blank, the Default Print Queue in the Job's System definition is used.

Print Form

This field specifies the OpenVMS print form which will be used when the Report is submitted to a print queue. If you leave this field blank, the default form of the Report's print queue will be used.

File Name

This field specifies a valid OpenVMS file name. The use of this field is determined by the template definitions. The default *JAMS* templates use this field as the base for a reports file name and any missing fields are replaced as follows:

- A missing device/directory is replaced with the specification from the Job's System definition.
- A missing filename is replaced with the Report I.D.
- A missing extension is replaced with the Jobs run occurrence number.

Number of Copies

This field specifies the number of copies of the report which should be printed. If you specify 0, the report is not sent to a print queue. If the value you specify is greater than 1, a /COPIES= qualifier is added to the print command which *JAMS* generates.

The JAMS Database

Job Definitions

Retention Days

This field specifies the number of days the report file should be retained. *JAMS* will create a SET FILE/RETENTION= command in the parsed .JOB file which will set the file's expiration date. *JAMS* does *not* delete these files when the expiration date has passed.

If you specify a value of 0 for retention days, the /DELETE qualifier will be added to the print command which *JAMS* generates.

Note: If a report is to be printed more than once, then you must specify a value greater than 0 here. Otherwise, the /DELETE qualifier will be added to all of the print commands, and the file could be deleted before it has printed at all locations.

PRINT Qualifiers

The text entered here is appended to the Reports OpenVMS print command when *JAMS* parses the Job file. The text entered here must consist of valid qualifiers for the OpenVMS print command and must begin with a slash.

1.3.10 Dependencies

Job dependencies are used to insure that all prerequisite processing has been performed before a Job can begin executing. A Job dependency can be based on:

- The completion of a local Job.
- The completion of a remote Job.
- The value of a *JAMS* Variable.

1.3.10.1 Job Completion Fields

The job completion fields are:

Depends on Job

This is the name of the Job which the current Job depends upon. This Job must have completed since the last time that the Since Job completed or within the range of time specified in the days and hours fields. This is a required field.

Completion Severity

The "Depends on Job" must complete with a OpenVMS severity level which is equal to or better than this severity. The possible values, in order from good to bad, are S (Success), I (Informational), W (Warning), E (Error) and F (Fatal or Severe). This is a required field.

Since Job

This is the name of a Job which is used as a reference point when determining if the “Depends on Job” has completed. Many times this will be the same as the current Job (i.e. JOBA must have completed since the last time I ran).

This is an optional field, if you leave this field blank, the Time Range fields are used to determine if the dependency has been satisfied.

Since Job Completion Severity

When evaluating the last time that the Since Job completed, *JAMS* requires that the Job completed with a severity which is equal to or better than the severity entered here. The possible values, in order of severity, are S (Success), I (Informational), W (Warning), E (Error) and F (Fatal or Severe).

Time Range

The time range fields (number of days and hours) let you define a dependency which is time based and usually less restrictive than a normal dependency.

These are optional fields. If you leave the Time Range fields zero, and leave the Since Job field blank, then whenever the current job is submitted, it will wait in the queue until the dependent job completes with a satisfactory status.

An example of when a time based dependency could be useful would be a situation where you have one update job and a number of reporting jobs. If you ran the update job at the end of every month, followed by the reporting jobs, you could define a dependency for each of the reporting jobs which requires that the update job completed within the past 28 days.

Always wait if queued

If you enter a “Y” in this field, when the Job is ready to run, *JAMS* will look to see if the “depends on” Job is scheduled or executing. If one or more of the “depends on” Jobs are found, the current Job will remain in a pending state until all occurrences of the “depends on” Job have completed.

1.3.10.2 Remote Job Completion Fields

The remote job completion fields are:

Remote Node

This is the DECnet node name of the remote node, or node group, where the “Depends on Job” runs. This field is present only for Remote Job Completion dependencies.

JAMS treats a VMScluster as a single unit. You can specify any node in the remote VMScluster which is running *JAMS* with networking enabled. The dependent job can run on any node in the remote VMScluster and the dependency will be satisfied.

The JAMS Database

Job Definitions

An entry into this field must be either a valid *JAMS* Node Definition or a valid *JAMS* Node Group Definition. You can use the **Find** key to list the valid Nodes and Node Groups.

Creating a dependency which refers to a Node Group is functionally equivalent to creating a dependency for every node in the Node Group. The primary advantage of Node Groups is that when the list of nodes in the group changes, you do not have to change all of your dependent jobs.

Depends on Job

This is the name of the Job which the current Job depends upon. This Job must have completed since the last time that the Since Job completed or within the range of time specified in the days and hours fields. This is a required field.

Since this Job resides on the remote node, it is not validated during data entry.

Completion Severity

The “Depends on Job” must complete with a OpenVMS severity level which is equal to or better than this severity. The possible values, in order from good to bad, are S (Success), I (Informational), W (Warning), E (Error) and F (Fatal or Severe). This is a required field.

Since Job

This is the name of a Job which is used as a reference point when determining if the “Depends on Job” has completed. Many times this will be the same as the current Job (i.e. JOBA must have completed since the last time I ran).

This is an optional field, if you leave this field blank, the Time Range fields are used to determine if the dependency has been satisfied.

Since Job Completion Severity

When evaluating the last time that the Since Job completed, *JAMS* requires that the Job completed with a severity which is equal to or better than the severity entered here. The possible values, in order of severity, are S (Success), I (Informational), W (Warning), E (Error) and F (Fatal or Severe).

Time Range

The time range fields (number of days and hours) let you define a dependency which is time based and usually less restrictive than a normal dependency.

These are optional fields. If you leave the Time Range fields zero, and leave the Since Job field blank, then whenever the current job is submitted, it will wait in the queue until the dependent job completes with a satisfactory status.

An example of when a time based dependency could be useful would be a situation where you have one update job and a number of reporting jobs. If you ran the update job at the end of every month, followed by the reporting jobs, you could define a dependency for each of the reporting jobs which requires that the update job completed within the past 28 days.

Always wait if queued

If you enter a “Y” in this field, when the Job is ready to run, *JAMS* will look to see if the “depends on” Job is scheduled or executing. If one or more of the “depends on” Jobs are found, the current Job will remain in a pending state until all occurrences of the “depends on” Job have completed.

1.3.10.3 Variable Value Fields

The variable value fields are:

Variable Name

This is the name of the *JAMS* Variable which this Job will depend upon.

Condition

This is the boolean operator or code used to define how the comparison will be performed. Press the Find key to display a list of the valid conditions.

Comparison Value

This is the value which the Variable will be compared to. This field has no meaning if the condition code is “Changes”.

1.3.11 Source Options

The *Source Options* option is used to specify the Job's source command file and how it will be processed. The fields on this form include:

Execution Method

The Execution Method specifies which of the defined Methods should be used to execute this Job. The following Methods are defined when *JAMS* is installed. You can also create your own Methods with the CREATE METHOD command.

SUBMIT	OpenVMS batch job with direct command procedure.
PARSED	OpenVMS batch job with parsed command procedure.
JACKET	OpenVMS batch job with jacketed command procedure.
DETACHED	OpenVMS detached process with direct command procedure.
DETPARSED	OpenVMS detached process with parsed command procedure.
NT	Windows NT Agent with direct command procedure.
NTPARSE	Windows NT Agent with parsed command procedure.
NTCMD	Windows NT Agent with single line command.
REXEC	Rexec server with single line command.

The JAMS Database

Job Definitions

Job Source

The Job Source is either the filename and extension of the command file associated with this Job or the command for execution methods which require a single line command.

The Directory specification for the command file is provided by the System Definition. If the location specified in the System Definition is a default location, then you can include a full file specification in the Job Source Field.

If left blank, the default filename is the Job Name. The default extension is obtained from the Execution Method.

1.3.12 View Source

The *View Source* option displays this Job's source command file in a scrollable window.

1.3.13 Edit Source

The *Edit Source* option activates a OpenVMS text editor and lets you edit or view this Job's source command file. The default editor is TPU but you can select from TPU, LSE and EDT by defining the logical name JAMS_EDITOR to be TPU, LSEEDIT or EDT. With TPU and LSE you can also include qualifiers in the logical name which change the editor. For example:

```
$ DEFINE JAMS_EDITOR "LSEEDIT/SECTION=MYSECTION/NOCOMMAND"
```

1.3.14 Precheck Job

The *Precheck Job* option is used to define this Job's Precheck Job. A Precheck Job is used to check for user defined prerequisites for this Job. When a Job which has a Precheck Job is submitted, it will remain in a pending state until the Precheck Job completes successfully. JAMS will automatically submit the Precheck Job when the main Job is ready to run. By ready to run we mean that the Job is not held or waiting for a specific time, all of the Job's Dependencies have been satisfied, and there are sufficient Resources available.

A Precheck Job has four options.

- 1 It can loop, waiting for the prerequisites.
- 2 It can exit with the status JAMS_TRYAGAIN and JAMS will resubmit the Precheck Job after the precheck interval has elapsed.
- 3 It can exit with a bad status and JAMS will perform normal notification actions and then resubmit the Precheck job after the precheck interval has elapsed.

- 4 It can exit with the status JAMS_xCANCELJOB, where x is S, I, W, E or F and denotes the severity of the status. JAMS will cancel the Job which caused the Precheck to run and the final status of the Job which caused the Precheck to run will be the status which the Precheck Job returned.

If the Precheck Job is parsed then the following predefined parameters are available:

JAMS_TRYAGAIN	The integer value of the status JAMS_TRYAGAIN.
JAMS_SCANCELJOB	The integer value of the status JAMS_SCANCELJOB.
JAMS_ICANCELJOB	The integer value of the status JAMS_ICANCELJOB.
JAMS_WCANCELJOB	The integer value of the status JAMS_WCANCELJOB.
JAMS_ECANCELJOB	The integer value of the status JAMS_ECANCELJOB.
JAMS_FCANCELJOB	The integer value of the status JAMS_FCANCELJOB.
JAMS_CANCELJOB	The integer value of the status JAMS_CANCELJOB. This status is also an E level status and is included for backwards compatibility.
JAMS_PRECHECK_ENTRY	The entry number of the Job which this Job is doing the checking for.
JAMS_PRECHECK_COUNT	The number of times that the Precheck Job has run for this entry.

1.3.15 Recovery Job

The *Recovery Job* option is used to define this Job's Recovery actions. The fields on this form include:

Recovery Job Severity

If this Job completes with a severity which is equal to or worse than the severity specified on this screen, the Recovery Job will be automatically submitted.

The Recovery Job is *not* submitted if this Job is intentionally aborted.

Recovery Job

This is the name of a valid JAMS Job which is to be submitted if this Job completes with a severity which is equal to or worse than the specified severity.

If the Recovery Job is parsed then the following predefined parameters are available:

JAMS_RECOVERY_ENTRY	The OpenVMS queue entry number of the Job that failed.
---------------------	--

The JAMS Database

Job Definitions

JAMS_RECOVERY_JAMS_ENTRY	The <i>JAMS</i> entry number of the Job that failed.
JAMS_RECOVERY_JOB_NAME	The <i>JAMS</i> Job name of the Job that failed.
JAMS_RECOVERY_JOB_STATUS	The most recent status text set by the failed Job using the <i>JAMS SET STATUS</i> command.
JAMS_RECOVERY_LOG_FILENAME	The full file specification of the log file for the job which failed.
JAMS_RECOVERY_MAIL_ADR	The list of names which will be notified via OpenVMS Mail.
JAMS_RECOVERY_OPER_CLASSES	The list of OPCOM classes which will be notified.
JAMS_RECOVERY_PID	The OpenVMS PID of the failed Job.
JAMS_RECOVERY_REPLY_USERS	The list of names which will receive a broadcast message.
JAMS_RECOVERY_RESTART_COUNT	The number of times the failed job has been restarted.
JAMS_RECOVERY_RON	The <i>JAMS</i> run occurrence number of the failed Job.
JAMS_RECOVERY_STATUS	The final OpenVMS status of the failed Job.
JAMS_RECOVERY_SYSTEM_ID	The <i>JAMS</i> System I.D. of the failed Job.

1.3.16 Recovery Instructions

The *Recovery Instructions* option is used to edit or view this Job's restart/recovery instructions. These instructions are included in the OpenVMS mail message notification messages.

You can also create restart/recovery instructions at the System level.

1.3.17 Notify Names

The *Notify Names* screen is used to define who should be notified when this Job requires notification.

These names can augment or replace the names specified on the Job's System definition. To remove a name which is specified in the System definition prefix the name with a minus sign. To remove all named specified in the System definition, enter “-”.

Notification is performed when:

- The Job completes with a severity which is worse than the Job's notification severity.
- The Job runs too long.
- The Job runs too quickly.
- The Job is stalled.

- The Job misses it's execution window.

The fields on this category include:

OpenVMS Mail Address

This is a list of OpenVMS Mail addresses, separated by commas. You can also specify logical names which equate to one or more OpenVMS Mail addresses.

OpenVMS REPLY Username List

This is a list of OpenVMS usernames, separated by commas. You can also specify logical names which equate to one or more OpenVMS usernames.

Operator Classes

This is a list of OpenVMS operator classes, separated by commas. You can also specify logical names which equate to one or more OpenVMS operator classes.

Valid OpenVMS operator classes are:

CARDS
CENTRAL
CLUSTER
DEVICES
DISKS
NETWORK
OPER1 through OPER12
PRINTER
SECURITY
TAPES

Notification Job Name

This is the name of a *JAMS* Job which should be submitted when notification is required. A Job specified here will override a Notification Job specified in the System definition.

If the Notification Job is parsed then the following predefined parameters are available:

JAMS_NOTIFY_ENTRY	The OpenVMS queue entry number of the job which caused the notification.
JAMS_NOTIFY_JAMS_ENTRY	The <i>JAMS</i> entry number of the job which caused the notification.
JAMS_NOTIFY_JOB_NAME	The Job Name of the job which caused the notification.
JAMS_NOTIFY_JOB_STATUS	The current Job Status value of the job which caused the notification.
JAMS_NOTIFY_LOG_FILENAME	The full file specification of the log file for the job which caused the notification.
JAMS_NOTIFY_MAIL_ADR	The list of names which will be notified via OpenVMS Mail.

The JAMS Database

Job Definitions

JAMS_NOTIFY_OPER_CLASSES	The list of OPCOM classes will receive an operator message.
JAMS_NOTIFY_PID	The process ID of the job which caused the notification.
JAMS_NOTIFY_REASON	The reason for the notifications. This parameter could be: COMPLETED, FAILED, SHORT, RUNAWAY, STALLED or MISSED_WINDOW.
JAMS_NOTIFY_REPLY_USERS	The list of names which will receive a broadcast message.
JAMS_NOTIFY_RON	The JAMS run occurrence number of the job which caused the notification.
JAMS_NOTIFY_STATUS	The final status of the job which caused the notification.
JAMS_NOTIFY_SUBMITTED_BY	The username which submitted the job which caused the notification.
JAMS_NOTIFY_SYSTEM_ID	The JAMS System I.D. of the job which caused the notification.

1.3.18 Notify Options

Notify options specified in a Job Definition override those specified in a System Definition.

Notify Severity

This field is used to define the minimum completion severity which will cause notification messages to be sent. If the Job's completion severity is equal to or worse than the specified severity, notification messages will be sent. You can also enter "N" to disable notification.

Stalled Job

The Stalled Time specifies how much time may elapse after the Job's scheduled time before the job is considered to be stalled. When the job is considered stalled, JAMS will perform notification for the job.

Runaway Job Elapsed Time

The Runaway Job Elapsed Time specifies how long the job may run before it is considered a runaway job. This is specified as a delta time. When the job is considered runaway, JAMS will perform notification for the job.

Runaway Job Elapsed Percent

The Runaway Job Elapsed Time Percent specifies how long the job may run before it is considered a runaway job. This is specified as a percentage of the Job's average elapsed time. When the job is considered runaway, JAMS will perform notification for the job.

Runaway Job CPU Time

The Runaway Job CPU Time specifies how much CPU time the job may consume before it is considered a runaway job. This is specified as a delta time. When the job is considered runaway, *JAMS* will perform notification for the job.

Runaway Job CPU Percent

The Runaway Job CPU Time Percent specifies how much CPU time the job may consume before it is considered a runaway job. This is specified as a percentage of the Job's average CPU time. When the job is considered runaway, *JAMS* will perform notification for the job.

Short Job Elapsed Time

The Short Job Elapsed Time specifies a minimum elapsed time for a job. If the job completes successfully in less than this amount of time it is considered a *Short Job*. When a job is determined to be a short job, *JAMS* will perform notification for the job and may change the completion severity of the job.

Short Job Elapsed Percent

The Short Job Elapsed Time Percent specifies a minimum elapsed time for a job. If the job completes successfully in less than this amount of time it is considered a *Short Job*. This is specified as a percentage of the Job's average elapsed time. When a job is determined to be a short job, *JAMS* will perform notification for the job and may change the completion severity of the job.

Short Job CPU Time

The Short Job CPU Time specifies a minimum CPU time for a job. If the job completes successfully but used less than this amount of CPU time it is considered a *Short Job*. When a job is determined to be a short job, *JAMS* will perform notification for the job and may change the completion severity of the job.

Short Job CPU Percent

The Short Job CPU Time Percent specifies a minimum CPU time for a job. If the job completes successfully but used less than this amount of CPU time it is considered a *Short Job*. This is specified as a percentage of the Job's average CPU time. When a job is determined to be a short job, *JAMS* will perform notification for the job and may change the completion severity of the job.

Short Job Completion Severity

If specified, when a job is determined to be a *Short Job*, the completion severity will be changed to the severity specified here.

1.3.19 Log Options

Log File options specified in a Job Definition override those specified in a System Definition.

Keep Logs

When the field is true, or “Y”, batch log files will not be deleted at the completion of the Job.

Print Logs

When the field is true, or “Y”, batch log files will be printed when the Job completes.

Time Stamp .LOG File Names

When the field is true, or “Y”, batch log files be time stamped with an extension of .LOG_yyyymmdd_hhmmsscc.

Log File Name

You can override the default filename for the log file created when this job runs.

1.3.20 Resources

The *Resources* option will display a list of the Resource Requirements which are currently defined for this Job.

Resource requirements specified in a Job Definition augment the Resource Requirements specified in the Job's System Definition. If the Job and System both specify the same Resource, the requirements are summed.

1.3.21 Execution Statistics

The *Execution Statistics* option displays performance statistics for this Job. These statistics are updated only if the Job completes with a severity level of Success or Informational. The screen shows the average, minimum and maximum values for the following statistics:

Count	The number of times that one or more fields in this column has been updated.
Elapsed Time	The total wall clock time from the start of execution to completion.
CPU Time	The amount of CPU time used. This is weighted according to the CPU rating established by the logical name JAMS_CPU_RATING.
Direct I/O	The number of direct I/O operations performed.
Buffered I/O	The number of buffered I/O operations performed.

Peak Working Set	The largest number of pages of memory used by this Job at any one time.
Page Faults	The total number of hard and soft page faults incurred by this Job.
Peak Virtual Pages	The largest number of pages of virtual memory used by this Job at any one time.

1.3.22 Completion Dates

The *Completion Dates* option shows the date and time of the last time this Job completed with each of the five possible completion severity levels.

If a date and time are all spaces, then the Job has never completed with that severity level.

1.3.23 References

The *References* screen lists all of the references to this Job. A Job may be referenced by:

- 1 Setup definitions
- 2 Trigger Actions
- 3 Trigger Events
- 4 Another Job's dependencies
- 5 As another Job's Recovery Job
- 6 As another Job's Precheck Job

1.4 Setup Definitions

Setup definitions are used to create a specific instance, or view, of a Job or sequence of Jobs. A *JAMS* Setup can set up a single job or a sequence of related jobs.

1.4.1 Setup Name

A Setup Name must be a valid OpenVMS identifier. This means that it must begin with an alphabetic character and contain only the characters A-Z, 0-9, \$ (dollar sign), and _ (underscore). Also the last character may not be an underscore.

1.4.2 Description

This is a description of this Setup definition. This description will be used on menus which are presented to a user who is selecting a batch job.

The description consists of two lines. If your description requires only one line, use the first line.

1.4.3 System

The System I.D. specifies to which System this Setup belongs. This is a required field and must be a System which has already been defined.

To create a Setup definition, you need to have ADD access to Setup definitions and you must have DEFINE_SETUP access to the System I.D. specified in this field.

1.4.4 Security

If you select the *Security* option and you have CONTROL access to the Setup (or the OpenVMS BYPASS privilege), the Setup's Access Control List will be displayed. You can then make modifications to the ACL. The following access rights can be specified:

Right	Meaning
CONTROL	Allows modification of this Setup's Access Control List.
CHANGE	Allows modification of this Setup definition provided that the user also has CHANGE access to Setup Definitions.
INQUIRE	Allows inquiry into this Setup definition provided that the user also has INQUIRE access to Setup Definitions.
DELETE	Allows deletion of this Setup definition provided that the user also has DELETE access to Setup Definitions.
SUBMIT	Allows submission of this Setup.
DEBUG	Allows submission of this Setup but only if the /DEBUG qualifier is used on the JAMS SUBMIT command. This qualifier will submit the Setup under the users OpenVMS username rather than the OpenVMS username specified in the System Definition.
OPERATOR	Allows the user to use the JAMS Job Monitor to hold, reschedule, release or delete an occurrence of this Setup.
ABORT_JOBS	Allows the user to use the JAMS Job Monitor to abort or restart an occurrence of this Setup.

1.4.5 Submit Options

The Submit Options form is used to define the options used when this Setup is submitted. The fields on this screen include:

Priority Modifier

The Scheduling Priority Modifier field is used to calculate the scheduling priority for this Setup. This number may range from -127 to 127, the default being 0. When a Setup is submitted by the JAMS submit sub-system, the scheduling priority is calculated by adding together the OpenVMS default scheduling priority, the System's priority modifier, the Job's priority modifier and the Setup's priority modifier.

Retain in Queue

Specifies the retention policy for this Setup. The options are:

Always	Setup is always retained until specifically deleted.
Never	Setup is never retained in the queue.
Error	Setup is retained if they complete with a severity of Warning or worse.
Timed	Setup is retained for the specified time after completion.
Not Specified	Use the default specified in the System Definition.

Hold Setup when submitted

This field specifies whether or not this Setup should be placed in a "holding" state when it is submitted.

This is useful for jobs which are requested, or submitted, by end-users but require operator approval before execution.

The JAMS Database

Setup Definitions

Suppress Display on Submit Menus

This field specifies whether or not this Setup should be excluded from *JAMS* Submit menus. If you specify “Yes” here, the Setup will never be displayed on a Submit menu. The only way to submit the Setup is via the *JAMS SUBMIT* command or automatically by specifying a scheduled date and time and a “Yes” in the Automatic submit field.

Notify Submitter

This field specifies whether or not the /NOTIFY qualifier is used when submitting this Setup. Note that when /NOTIFY is specified, the username who submitted the job is notified, not the username under which the Setup ran.

Submit Under Username

You can specify the OpenVMS username under which this Setup should run. This will override the OpenVMS Username specified in the Setup's System Definition. You must have either the CMKRNL privilege or SUBMIT access to the Username which you specify. See the *JAMS SET USERNAME* command for information on setting access to Usernames.

1.4.6 Submit Times

Submit Window Time

This field is used to specify the name of a *JAMS* Named Time which defines this Setup's Submit Window. You can specify a Named Time or a specific time range but not both.

Setup may be scheduled between

These fields are used to specify a specific Submit Window for this Setup. You can specify a Named Time or a specific time range but not both.

Missed Window Action

This field specifies the action that *JAMS* should take if this Setup misses it's time window. Possible actions are:

No Action	The window is only used to limit the time which can be entered when the Setup is manually submitted.
Continue	The Setup will not start until the time window opens but will not be rescheduled or aborted if the window closes before the Setup completes.
Abort or Delete	The Setup will not start until the time window opens and will be deleted or aborted is the window closes before the Setup completes.

Reschedule or Continue	The Setup will not start until the time window opens. If the Setup has not started when the window closes, it will be rescheduled for the next time that the window opens. If it has started, it will be allowed to continue.
Restart or Reschedule	The Setup will not start until the time window opens. If the Setup has not complete when the window closes, it will be rescheduled for the next time the window opens. If the Setup is executing when the window closes it will be aborted and rescheduled.

Notify of Missed Window

Defines whether or not *JAMS* should perform notification if the Setup misses it's time window.

Default Submit Date

This field is used to specify a default submit date which is used when this Setup is manually submitted.

Default Submit Time

This field is used to specify a default submit time which is used when this Setup is manually submitted.

1.4.7 Schedule Options

Schedule Options are used by *JAMS* to automatically submit a Setup to run unattended.

Scheduled Date

This is an English language representation of the date, or dates, on which this Setup is scheduled to run.

You can specify multiple scheduled days by separating the text with commas. For example, "MONDAY,WED,FRIDAY" means that this Setup should be run every Monday, Wednesday and Friday.

Note that the quotations shown in the examples should not be entered in the scheduled date field. For a complete description of English language date representations, refer to Chapter 7 of the *JAMS User Guide*.

Except for

This is an English language representation of a date, or dates, on which this Setup should *not* be run. If you want to schedule a Setup to run every Monday, except for the last Monday of the month, you would enter "MONDAY" into the *Scheduled Date* field and "LAST MONDAY OF MONTH" into this field.

Scheduled Time

This is the time of day when this Setup should be run.

The JAMS Database

Setup Definitions

Every hh:mm Minutes Until

If you want this Setup to run many times during a day, you can specify the delta time between runs and the time of day to stop repeating. Then, any time this Setup completes it will be resubmitted to run *hh:mm* later. The Setup will be resubmitted no matter what its completion status was or how it was submitted. The only way to stop this repetition is to intentionally delete the Setup from the schedule.

You can specify what time to use as the base for calculating the next run time, the base may be the Scheduled, Start or End time.

Automatic Submit at Scheduled Date and Time

This is a Y (yes) or N (no) field and indicates whether or not this Setup should be automatically submitted at the scheduled date and time.

A N (no) entry indicates that the scheduled date and time are to be used only for management and capacity planning.

Non-workday Scheduling

This field indicates how this Setup should be scheduled if the day it is scheduled to run falls on a non-workday. There are three options:

- S - Schedule the Setup, even on non-workdays.
- I - Ignore the Setup, do NOT schedule on non-workdays.
- D - Defer the Setup until the next workday.

1.4.8 Jobs

A Setup can run an unlimited number of Jobs. The Jobs can be any mix of Execution Methods. A single Setup can run Jobs on OpenVMS, Windows NT and UNIX. These Jobs can run simultaneously, or one right after the other.

Job/Description

This is where you specify the individual *JAMS* Jobs to run within this Setup.

Job Name

Specifies the Job Name.

Step

Defines the Job's Step within the Setup. The Jobs in a Setup run in step order. All of the Jobs in the lowest numbered step are released. When all of the jobs in a step are complete the Setup will advance to the next higher step and release the Jobs in that step. This behavior can be modified with the *Wait for Completion of this Job before advancing Step* field.

Minimum Completion Severity for this Job

Specifies the minimum completion severity for this Job. For a Setup to be considered successful, all of the Jobs in the Setup must complete with a severity which meets their minimum.

Wait for Completion of this Job before advancing Step

When set to true, the Setup will wait for the completion of this job before advancing to the next step. When set to false, the Setup will advance to the next step even though this Job hasn't completed.

Let the Setup Continue if the Job fails

When set to true, if this Job fails the Setup will continue with the next Job in this Setup.

When set to false, if this Job fails the Setup will be halted. No further Jobs will be released. Jobs already released will continue. Manual intervention is required.

Override Name

Used to specify an override name for this Job.

Override Description

Used to specify an override description for this Job.

Submit Times

You can override the Submit Times which are specified in the Job Definition.

Schedule Options

Each Job in a Setup can have its own schedule. The Job will execute only when the Setup and Job schedules intersect. The default is to have the same schedule as the Setup so any time the Setup runs, the Job is included in the run.

Submit Options

You can override the Submit Options which are specified in the Job Definition.

Parameters

A Setup definition also consists of default values for a Job's parameters. You can also modify the attributes of a parameter.

When you select the *Parameters* option, a list which contains the Job's parameters is displayed. An asterisk next to a Parameter signifies that some of the Parameter's attributes are overridden in this Setup.

A Setup's Parameter definitions are very important when the Setup will be automatically submitted because the default values specified here (or in the Job definition) will be used when the Setup is submitted.

When you select a Parameter, a detail screen will be displayed which shows all of the Parameter fields which may be overridden in a Setup definition. The Job's values are shown in parentheses. If you leave a field blank, the value will be inherited from the Job.

The JAMS Database

Setup Definitions

In addition to overriding the defaults from the Job definition, you can specify a Global Name for the Parameter. All parameters in this Setup which have the same Global Name and datatype are combined into a single parameter.

Reports

A Setup definition also consists of default values for a Job's Reports.

When you select the *Reports* option, a list which contains the Job's Reports is displayed. An asterisk next to a Report signifies that some of the Report's attributes are overridden in this Setup.

A Setup's Report definitions are very important when the Setup will be automatically submitted because the default values specified here (or in the Job definition) will be used when the Setup is submitted.

When you select a Report, a detail screen will be displayed which shows all of the Report fields which may be overridden in a Setup definition. The Job's values are shown in parentheses. If you leave a field blank, the value will be inherited from the Job.

View Source

Displays the Jobs source code.

Precheck Job

You can override the Precheck Job which is specified in the Job Definition.

Recovery Job

You can override the Recovery Job which is specified in the Job Definition.

Notify Names

You can override the Notify Names which are specified in the Job Definition.

Notify Options

You can override the Notify Options which are specified in the Job Definition.

Execution Statistics

Displays the execution statistics from this Job when it is executed as a part of this Setup.

1.4.9 Resources

The *Resources* option will display a list of the Resource Requirements which are currently defined for this Setup.

Resource requirements specified in a Setup Definition augment the Resource Requirements specified in the Setup's System Definition. If the Setup and System both specify the same Resource, the requirements are summed.

1.4.10 Dependencies

The *Dependencies* option is used to define dependencies for this Setup.
For more information on dependency definitions, refer to Section 1.3.10.

1.4.11 Precheck Job

The *Precheck Job* option is used to define this Setup's Precheck Job. A Precheck Job is used to check for user defined prerequisites for this Job. When a Setup which has a Precheck Job is submitted, it will remain in a pending state until the Precheck Job completes successfully. JAMS will automatically submit the Precheck Job when the main Job is ready to run. By ready to run we mean that the Setup is not held or waiting for a specific time, all of the Setup's Dependencies have been satisfied, and there are sufficient Resources available.

A Precheck Job has four options.

- 1 It can loop, waiting for the prerequisites.
- 2 It can exit with the status JAMS_TRYAGAIN and JAMS will resubmit the Precheck Job after the precheck interval has elapsed.
- 3 It can exit with a bad status and JAMS will perform normal notification actions and then resubmit the Precheck job after the precheck interval has elapsed.
- 4 It can exit with the status JAMS_xCANCELJOB, where x is S, I, W, E or F and denotes the severity of the status. JAMS will cancel the Job which caused the Precheck to run and the final status of the Job which caused the Precheck to run will be the status which the Precheck Job returned.

If the Precheck Job is parsed or jacketed then the following predefined parameters are available, and defined as DCL symbols:

JAMS_TRYAGAIN	The integer value of the status JAMS_TRYAGAIN.
JAMS_SCANCELJOB	The integer value of the status JAMS_SCANCELJOB.
JAMS_ICANCELJOB	The integer value of the status JAMS_ICANCELJOB.
JAMS_WCANCELJOB	The integer value of the status JAMS_WCANCELJOB.
JAMS_ECANCELJOB	The integer value of the status JAMS_ECANCELJOB.
JAMS_FCANCELJOB	The integer value of the status JAMS_FCANCELJOB.
JAMS_CANCELJOB	The integer value of the status JAMS_CANCELJOB. This status is also an E level status and is included for backwards compatibility.

The JAMS Database

Setup Definitions

JAMS_PRECHECK_ENTRY	The entry number of the Job which this Job is doing the checking for.
JAMS_PRECHECK_COUNT	The number of times that the Precheck Job has run for this entry.

1.4.12 Recovery Job

The *Recovery Job* option is used to define this Setup's Recovery actions. The fields on this form include:

Recovery Job Severity

If this Setup completes with a severity which is equal to or worse than the severity specified on this screen, the Recovery Job will be automatically submitted.

The Recovery Job is *not* submitted if this Job is intentionally aborted.

Recovery Job

This is the name of a valid *JAMS* Job which is to be submitted if this Setup completes with a severity which is equal to or worse than the specified severity.

If the Recovery Job is parsed then the following predefined parameters are available:

JAMS_RECOVERY_ENTRY	The OpenVMS queue entry number of the Job that failed.
JAMS_RECOVERY_JAMS_ENTRY	The <i>JAMS</i> entry number of the Job that failed.
JAMS_RECOVERY_JOB_NAME	The <i>JAMS</i> Job name of the Job that failed.
JAMS_RECOVERY_JOB_STATUS	The most recent status text set by the failed Job using the <i>JAMS SET STATUS</i> command.
JAMS_RECOVERY_LOG_FILENAME	The full file specification of the log file for the job which failed.
JAMS_RECOVERY_MAIL_ADR	The list of names which will be notified via OpenVMS Mail.
JAMS_RECOVERY_OPER_CLASSES	The list of OPCOM classes which will be notified.
JAMS_RECOVERY_PID	The OpenVMS PID of the failed Job.
JAMS_RECOVERY_REPLY_USERS	The list of names which will receive a broadcast message.
JAMS_RECOVERY_RESTART_COUNT	The number of times the failed job has been restarted.
JAMS_RECOVERY_RON	The <i>JAMS</i> run occurrence number of the failed Job.
JAMS_RECOVERY_STATUS	The final OpenVMS status of the failed Job.

JAMS_RECOVERY_SYSTEM_ID

The JAMS System I.D. of the failed Job.

1.4.13 Recovery Instructions

The *Recovery Instructions* option is used to edit or view this Setup's restart/recovery instructions. These instructions are included in the OpenVMS mail message notification messages.

You can also create restart/recovery instructions at the Job and System level.

1.4.14 Notify Names

The *Notify Names* screen is used to define who should be notified when this Setup requires notification.

These names can augment or replace the names specified on the Job and System definitions. To remove a name which is specified in the Job or System definition prefix the name with a minus sign. To remove all previous names, enter "-*".

Notification is performed when:

- The Setup completes with a severity which is worse than the Setup's notification severity.
- The Setup runs too long.
- The Setup runs too quickly.
- The Setup is stalled.
- The Setup misses its execution window.

The fields on this category include:

OpenVMS Mail Address

This is a list of OpenVMS Mail addresses, separated by commas. You can also specify logical names which equate to one or more OpenVMS Mail addresses.

OpenVMS REPLY Username List

This is a list of OpenVMS usernames, separated by commas. You can also specify logical names which equate to one or more OpenVMS usernames.

Operator Classes

This is a list of OpenVMS operator classes, separated by commas. You can also specify logical names which equate to one or more OpenVMS operator classes.

Valid OpenVMS operator classes are:

CARDS
CENTRAL
CLUSTER

The JAMS Database

Setup Definitions

DEVICES
DISKS
NETWORK
OPER1 through OPER12
PRINTER
SECURITY
TAPES

Notification Job Name

This is the name of a *JAMS* Job which should be submitted when notification is required. A Job specified here will override a Job specified in the System definition.

If the Notification Job is parsed then the following predefined parameters are available:

JAMS_NOTIFY_ENTRY	The OpenVMS queue entry number of the job which caused the notification.
JAMS_NOTIFY_JAMS_ENTRY	The <i>JAMS</i> entry number of the job which caused the notification.
JAMS_NOTIFY_JOB_NAME	The Job Name of the job which caused the notification.
JAMS_NOTIFY_JOB_STATUS	The current Job Status value of the job which caused the notification.
JAMS_NOTIFY_LOG_FILENAME	The full file specification of the log file for the job which caused the notification.
JAMS_NOTIFY_MAIL_ADR	The list of names which will be notified via OpenVMS Mail.
JAMS_NOTIFY_OPER_CLASSES	The list of OPCOM classes will receive an operator message.
JAMS_NOTIFY_PID	The process ID of the job which caused the notification.
JAMS_NOTIFY_REASON	The reason for the notifications. This parameter could be: COMPLETED, FAILED, SHORT, RUNAWAY, STALLED or MISSED_WINDOW.
JAMS_NOTIFY_REPLY_USERS	The list of names which will receive a broadcast message.
JAMS_NOTIFY_RON	The <i>JAMS</i> run occurrence number of the job which caused the notification.
JAMS_NOTIFY_STATUS	The final status of the job which caused the notification.
JAMS_NOTIFY_SUBMITTED_BY	The username which submitted the job which caused the notification.
JAMS_NOTIFY_SYSTEM_ID	The <i>JAMS</i> System I.D. of the job which caused the notification.

1.4.15 Notify Options

Notify options specified in a Setup Definition override those specified in a System Definition.

Notify Severity

This field is used to define the minimum completion severity which will cause notification messages to be sent. Setups only have two completion severities, Success and Failure. This field must be "S", "F" or "N" (Never).

Stalled Job

The Stalled Time specifies how much time may elapse after the Setup's scheduled time before the it is considered to be stalled. When the Setup is considered stalled, *JAMS* will perform notification.

Runaway Setup Elapsed Time

The Runaway Setup Elapsed Time specifies how long the Setup may run before it is considered a runaway Setup. This is specified as a delta time. When the Setup is considered runaway, *JAMS* will perform notification.

Runaway Setup Elapsed Percent

The Runaway Setup Elapsed Time Percent specifies how long the Setup may run before it is considered a runaway Setup. This is specified as a percentage of the Setup's average elapsed time. When the Setup is considered runaway, *JAMS* will perform notification.

Short Setup Elapsed Time

The Short Setup Elapsed Time specifies a minimum elapsed time for a Setup. If the Setup completes successfully in less than this amount of time it is considered a *Short Setup*. When a Setup is determined to be a short Setup, *JAMS* will perform notification for the Setup and may change the completion severity of the Setup.

Short Setup Elapsed Percent

The Short Setup Elapsed Time Percent specifies a minimum elapsed time for a Setup as a percentage of it's average elapsed time.

Short Setup Completion Severity

If specified, when a Setup is determined to be a *Short Setup*, the completion severity will be changed to the severity specified here.

1.4.16 Log Options

Log File options specified in a Setup Definition override those specified in a System Definition.

The JAMS Database

Setup Definitions

Keep Logs

When the field is true, or “Y”, batch log files will not be deleted at the completion of the Setup.

Print Logs

When the field is true, or “Y”, batch log files will be printed when the Setup completes.

Time Stamp .LOG File Names

When the field is true, or “Y”, batch log files be time stamped with an extension of .LOG_yyyymmdd_hhmmsscc.

Log File Name

You can override the default filename for the log file created when this Setup runs.

1.4.17 Execution Statistics

The *Execution Statistics* option displays performance statistics for this Setup. These statistics are updated only if the Setup completes with a severity level of Success or Informational. The screen shows the average, minimum and maximum values for the following statistics:

Count	The number of times that one or more fields in this column has been updated.
Elapsed Time	The total wall clock time from the start of execution to completion.
CPU Time	The amount of CPU time used. This is weighted according to the CPU rating established by the logical name JAMS_CPU_RATING.
Direct I/O	The number of direct I/O operations performed.
Buffered I/O	The number of buffered I/O operations performed.
Peak Working Set	The largest number of pages of memory used by this Job at any one time.
Page Faults	The total number of hard and soft page faults incurred by this Job.
Peak Virtual Pages	The largest number of pages of virtual memory used by this Job at any one time.

1.4.18 Completion Dates

The *Completion Dates* option shows the date and time of the last time this Job completed with each of the five possible completion severity levels.

If a date and time are all spaces, then the Job has never completed with that severity level.

Also shown is the date and time of the last time that this Setup was automatically submitted by the JAMS_AUTOSUBMIT job.

1.5 Trigger Definitions

Triggers are used to define event based schedules. A Trigger consists of a list of events and a list of actions. When *JAMS* sees that all of a Trigger's events are true, it will perform the list of actions. With this version of *JAMS*, the only supported action is to submit a job.

Triggers can be maintained with the screen or Motif based interfaces.

Trigger Name

The Trigger Name is a unique identifier for a Trigger.

A Trigger Name must be a valid OpenVMS identifier. This means that it must begin with an alphabetic character and contain only the characters A-Z, 0-9, \$ (dollar sign), and _ (underscore). Also the last character may not be an underscore.

Description

The description is used in lists and reports to provide a more complete description of the Trigger.

Reset When Fired

This is a Y (yes) or N (no) field which specifies whether or not this Trigger should be reset when it is fired. When a Trigger is fired, it will not be fired again until it has been reset. If you enter a N into this field, the Trigger will not be reset when it is fired and must be manually reset with the RESET TRIGGER command.

Status

This is a display only field which shows the current status of the Trigger. Possible values are:

Enabled	The Trigger is enabled and ready to be fired.
Disabled	The Trigger has been disabled with the DISABLE TRIGGER command. It will not be fired.
Fired	The Trigger has been fired and has not yet been reset.

Last Fired

This is a display only field which shows the date and time that this Trigger was last fired.

Last Reset

This is a display only field which shows the date and time that this Trigger was last reset.

The JAMS Database

Trigger Definitions

1.5.1 Events

There are two types of Events: Job Completion Events and Variable Value Events.

A Job Completion Event is true if the specified Job has completed with the specified severity level since the last time that the Trigger was reset.

A Variable Value Event is true or false based strictly upon the value of the specified variable. There is no relation to the Trigger's last reset time.

Note that the order of events is not important.

1.5.1.1 Job Completion Events

Job Name

This is the name of the *JAMS* Job which this event will depend upon.

Severity

This is the minimum, or maximum, severity needed to cause this event to be true.

Note that if you intentionally delete a job from the schedule, it will not cause a Trigger event to be true.

Better or Worse

This field must be either B (better) or W (worse) and is used in conjunction with the Severity field to determine if an instance of the specified Job should cause this event to be true.

1.5.1.2 Variable Value Event Fields

Variable Name

This is the name of the *JAMS* Variable which this event will depend upon.

Condition

This is the boolean operator or code used to define how the comparison will be performed. Press the Find key to display a list of the valid conditions.

Comparison

This is the value which the Variable will be compared to. This field has no meaning if the condition code is "Changes".

1.5.2 Actions

Each Trigger Action is a Job which will be submitted when the Trigger fires.

Job Name

This is the name of a Job which will be submitted when this Trigger is fired. It must be a valid Job and you must have SUBMIT access to the Job.

Submit Date/Submit Time

You can use the Submit Date and Submit Time fields to specify when this Job should be scheduled to run, once the Trigger fires.

Submit on Hold

This field must be either Y (yes) or N (no) and specifies whether or not this Job should be placed in a “holding” state when it is submitted.

OpenVMS Username

The OpenVMS Username under which this Job will be submitted is shown in a display only field. If this Job's System I.D. specifies a OpenVMS Username, it is used. Otherwise the Username is the Username of the last person to modify this Trigger.

1.6 Variable Definitions

Variables are used to hold single pieces of data which may be used in your batch scheduling and/or processing. A *JAMS* Variable is a piece of data which can be used:

- as a default value for a Job's parameter.
- as an undefined parameter in a parsed Job.
- in Job Dependencies.
- in Trigger Events.
- in your application programs and/or DCL command procedures.

Variables can be maintained with the screen, command or Motif based interfaces.

Variable Name

The Variable Name is a unique identifier for a Variable. You can reference a Variable in a *JAMS* parsed Job by specifying the Variable Name enclosed in double <> signs (i.e. <<variable_name>>). When the Job is submitted, the <<variable_name>> will be replaced with the current value of the Variable.

JAMS also provides commands and routines which allow you to get and/or set the value of a Variable from DCL or a 3GL language.

A Variable Name must be a valid OpenVMS identifier. This means that it must begin with an alphabetic character and contain only the characters A-Z, 0-9, \$ (dollar sign), and _ (underscore). Also, the last character may not be an underscore.

Description

This field is used to more fully describe this Variable.

Data Type

You must specify the data type of a Variable. Possible values are:

TEXT	Up to 256 bytes of text.
INTEGER	A longword integer value.
BOOLEAN	A true or false value.
DATE	A date.
TIME	A time of day.
DATETIME	A date and time of day.
FLOAT	A floating point number.
UNKNOWN	Any data type. <i>JAMS</i> cannot test the value of an unknown Variable other than to see when it changes. You cannot change the value of an unknown Variable with the <i>JAMS</i> full screen interface.

Value

This is the current value of the Variable.

Security

If you are properly authorized, you can view or edit a Variable's Access Control List (ACL). This ACL specifies who may see and/or modify the value of this Variable.

1.7 Menu Definitions

Menu Definitions define customized menu hierarchies for the *JAMS* Submit Menu. These menus are used by *JAMS* to present a hierarchy of menus to a user when they want to submit a batch Job. These menus are available from the *Job Submission* menu option within the screen based interface, the MOTIF interface, and the Windows 9x/NT/2000 interface. You can also access these menus from within your application via the *JAMS* callable interface.

Creating menus is optional. *JAMS* can create a default menu which initially lists all of the available Systems. After the user selects a System, all of the available Jobs and Setup definitions are then displayed.

Menus can be maintained with the screen or Motif based interfaces.

Menu Name

This is a unique identifier for this menu.

A Menu Name must be a valid OpenVMS identifier. This means that it must begin with an alphabetic character and contain only the characters A-Z, 0-9, \$ (dollar sign), and _ (underscore). Also the last character may not be an underscore.

Description

The description will be used when a menu is built. The Menu Name and menu description are displayed as part of the menu.

1.7.1 Menu Detail

The Menu definition form also contains a list of the Menu's detail lines. Each detail line is edited independently.

A single menu definition may have many detail lines. Each detail line specifies selection criteria used to build a new menu when this Menu Name is specified. Each detail line is additive. The Menus, Jobs and/or Setups selected by a detail lines selection criteria are added to the ones selected by previous detail lines to build a complete menu.

System I.D.

If you enter a value for the System I.D., only Jobs or Setups from the indicated System are included in the menu.

Search Specification

Only Jobs, Setups and/or Menus which match this specification are included in the menu. You can use standard OpenVMS wildcards in this specification.

Include Jobs

If you specify Y (yes), Jobs which match the selection criteria will be included in the menu. If you specify N (no), Jobs will never be included in the Menu.

Include Setups

If you specify Y (yes), Setups which match the selection criteria will be included in the menu. If you specify N (no), Setups will never be included in the Menu.

Include Menus

If you specify Y (yes), Menus which match the selection criteria will be included in the menu. If you specify N (no), Menus will never be included in the Menu.

1.8 Named Times

Named Times are used to provide flexible windows of time when Job's may be scheduled. *JAMS* can prevent Job's from starting until their Named Time window is enabled and can take various actions when a Job's window closes before the Job completes.

Named Times can be maintained with the command, screen or Motif based interfaces. They can be enabled and disabled with the command based interface.

Time Name

This field is a unique identifier for this Named Time.

A Time Name must be a valid OpenVMS identifier. This means that it must begin with an alphabetic character and contain only the characters A-Z, 0-9, \$ (dollar sign), and _ (underscore). Also the last character may not be an underscore.

Description

The description is used in menus, lists and reports to provide a more complete description of the Time than provided by the Named Times identifier.

Start Time

This is the time of day when the Named Time is scheduled to be enabled.

End Time

This is the time of day when the Named Time is scheduled to be disabled.

Automatic Enable

This field specifies whether or not the Named Time should be automatically enabled. If not automatically enabled, the Named Time will remain disabled until the *ENABLE TIME time-name* command is used to manually enable the Named Time.

Automatic Disable

This field specifies whether or not the Named Time should be automatically disabled. If not automatically disabled, the Named Time will remain enabled until the *DISABLE TIME time-name* command is used to manually disable the Named Time.

1.9 Date Types

Date Types are high level definitions of special dates, such as holidays or fiscal periods. Date Types are used to define the various classes of dates which are significant to your environment. This is not where you define the specific dates for these classes. Rather it is where you define the classes themselves. The specific dates are defined using the Date Definitions menu option.

You must be careful when you choose the identifiers for your Date Types and Specific Date Types. These names will be used in English language date specifications so they should be readable names. *JAMS* also recognizes month names before checking for Date Types so you cannot use Date Type definitions to override the calendar months.

For example, *JAMS* will always convert "FIRST DAY OF APRIL" to April 1st of the current calendar year. If your company uses fiscal accounting periods, your accountants may have some other date in mind when they think of the first day of April. You can define Date Types and Dates to deal with this problem.

If you define a Date Type for fiscal periods and give it the identifier FISCAL with the names of months as Specific Date Types, your accountants can specify "FIRST DAY OF FISCAL APRIL" to specify the first day of the April fiscal period. They could also specify "FIRST DAY OF FISCAL" to specify the first day of the current fiscal period.

Note that there is nothing special about the name FISCAL. It was chosen to make the date text more readable and could just as easily have been PERIOD or FP.

You may want to use unique names for the Specific Date Types in a Date Type definition. In the previous example, we could have used FP_JAN, FP_FEB etc. for the specific date type names. Then you could express the first day of fiscal april as "FIRST DAY OF FP_APR".

Date Type

This field is a unique identifier for this Date Type. It will be used in English language date definitions so it should also be a fairly descriptive identifier.

Description

The description is used in menus, lists and reports to provide a more complete description of the Date Type than provided by the Date Type's identifier.

Continuous

The Continuous field is a Yes/No field which indicates whether or not this Date Type is continuously occurring. A continuous Date Type is one which spans a number of dates and consequently, is continuously occurring.

The JAMS Database

Date Types

The most common example of a noncontinuous Date Type would be Holidays. Your environment may have other situations where a noncontinuous Date Type would be useful. One example would be a Company which takes physical inventories. If you have special batch processing which should be run on a day when physical inventory is taken, you could create a Date Type of PHYSICAL and then define Setups which are scheduled to run on PHYSICAL. When a physical inventory is scheduled, you simply need to add the dates and the jobs will be correctly scheduled.

You could also use the PHYSICAL Date Type to obtain default values for a jobs parameters. You could specify "LAST PHYSICAL" as the default value for a date parameter.

The most common example of a continuous Date Type would be a fiscal period. To define your company's fiscal periods, create a Date Type, such as FISCAL, and then define the starting date of each fiscal period. You may need additional continuous Date Types for periods such as fiscal quarters, pay periods etc.

Valid Specific Date Types

You have 24 fields in which you can specify the names of specific occurrences of this Date Type. The order of these names does not matter except for the first name. The first specific date type must identify the date or period which occurs first in any given year. The concept of "Year" is user defined. You can specify the first date/period to occur in a calendar year or a fiscal year.

1.10 Date Definitions

Date Definitions are used to mark dates which are significant to your batch processing. These dates may be company observed holidays, fiscal periods, etc. You should be careful when defining these dates since they are used to schedule (or not schedule) Jobs, and they are used to create the defaults for dates used as parameters to Jobs. If your Data Center operates 365 days a year and schedules jobs based on calendar months, you do not need to define any special dates.

There are two types of significant dates. One type marks a single day as having some importance. The second type marks the beginning of a period. If the *Date Type* of the date you are defining is a continuous Date Type, then any dates which you define mark the beginning of a new period. If the Date Type is not continuous, then the date marks a single significant date. Refer to Section 1.9 for information on defining Date Types.

Date Type

This identifies the Date Type of the date which you are defining. Every Date definition must be associated with a Date Type, but the same date may be associated with more than one Date Type.

Date Types are user defined. Refer to the Configuration menu option for information on defining Date Types.

Starting Date

This is the date you are defining. If the Date Type you specified is a continuous Date Type, then this date represents the first day of a period, otherwise it represents a single significant date.

Specific Type

This field represents the specific name of the date you are defining. This field is not required. However, if you specify a value, it must be one of the values defined in the Date Type's definition.

This field is significant when you want to refer to a specific instance of a Date Type. If you want to schedule a job to run on Christmas, you could define a Date Type of HOLIDAY with a Specific Date Type of CHRISTMAS.

For a continuous Date Type, such as a fiscal period, this field can be used to name each period. Generally fiscal periods correspond to calendar months so you could use Specific Date Types such as JAN, FEB, MAR etc. This would allow you to enter a date specification such as "2ND MONDAY OF FISCAL APR".

Description

The description should be used to accurately identify this date definition. It is for reference purposes only.

The JAMS Database

Date Definitions

Workday

You have three options for this field: Y (yes), N (no), or space which means “maybe”. When *JAMS* needs to know if a date is a workday, first it looks at the date definitions for the date in question. If it finds one with either a Y or an N in this field, it stops checking and has the answer. (Y means yes this is a workday, N means no this is not a workday). If none of the date definitions for this date specify Y or N (or there are no definitions for this date), then *JAMS* checks the configuration to see if the day of the week on which this date falls is normally a workday.

Normally, you should leave this field blank unless one of the reasons for creating the date definition is to change the workday status. Be careful because if you create two definitions for the same date, one that says Yes this is a workday and one that says No this is not a workday, *JAMS* will stop checking when it finds the first definition.

1.11 Configuration

The Configuration menu option is used to define and maintain many of the *JAMS* system wide options and configuration parameters. Generally, you will need to review and update these options when you first install *JAMS*, but you usually will not want to change these definitions frequently.

Normal Workdays

You must specify which days of the week are normally considered workdays. This workday status can be overridden by a specific date definition.

OpenVMS Mail Address

This is a list of OpenVMS Mail addresses, separated by commas. If an unregistered Job terminates with a bad status, a OpenVMS mail message will be sent to this list of addresses. The message will specify the Job which completed abnormally and provide the final status code and message text. If left blank, no mail message is sent.

You can also specify a list of logical names which translate into lists of OpenVMS Mail Addresses and/or logical names.

OpenVMS REPLY Username List

This is a list of OpenVMS usernames, separated by commas. If an unregistered Job terminates with a bad status, a message will be broadcast to these users (if they are logged on at the time). The message will specify the Job which completed abnormally and provide the final status code and message text. If left blank, no broadcast message is sent.

You can also specify a list of logical names which translate into lists of OpenVMS Usernames and/or logical names.

Operator Classes

This is a list of OpenVMS operator classes, separated by commas. If an unregistered Job terminates with a bad status, a message will be sent via OPCOM to these operator classes. The message will specify the Job which completed abnormally and provide the final status code and message text. If left blank, no operator message is sent.

Valid OpenVMS operator classes are:

- CARDS
- CENTRAL
- CLUSTER
- DEVICES
- DISKS
- NETWORK
- OPER1 through OPER12
- PRINTER
- SECURITY
- TAPES

The JAMS Database

Configuration

You can also specify a list of logical names which translate into lists of Operator Classes and/or logical names.

Directory for Temporary Job Files

JAMS has the ability to parse a command file template and create a temporary command file which is submitted as a batch job. This temporary command file is deleted when the batch job completes. Specify a device and directory, or a logical name which expands to a device and directory, in this field.

This directory should be a secure directory. The UIC of the owner should be a trusted UIC (such as [1,1]) and the protection on the directory should be set to: (S:RWE, O:RE, G:RE, W:RE)

Note: Any logical names referenced here must be defined at the executive level. Make sure that you use the /EXECUTIVE qualifier when you define the logical name.

Retain History from unregistered Jobs

The *JAMS* monitoring system monitors all batch jobs which run on an OpenVMS system. An unregistered Job is one which has not been defined in the *JAMS* database. If you set the Retain History from unregistered Jobs field to Y (Yes), a history record will be maintained for all unregistered Jobs. Set the switch to N (No) to ignore unregistered jobs.

Manage unknown jobs as if they were submitted by JAMS?

This switch defines whether *JAMS* will attempt to make an unknown job wait for dependencies. An unknown job is a job which is registered in the *JAMS* database but was not submitted by *JAMS*. If you set this switch to "Y", *JAMS* will check a Job's dependencies when it finds an unknown, but registered, Job pending in a batch queue. You must make sure that the job will remain in a pending state until *JAMS* sees the job in the queue. One way to do this is to include the qualifier /CHAR=JAMS_SCHEDULE on the SUBMIT command.

Treat unknown jobs as if they were submitted with /DEBUG?

This switch defines whether *JAMS* will treat unknown jobs as if they had been submitted with the *JAMS* /DEBUG qualifier. An unknown job is a job which is registered in the *JAMS* database but was not submitted by *JAMS*. Jobs which are submitted with the /DEBUG qualifier will not satisfy dependencies or cause Triggers to fire.

When *JAMS* discovers a job which it did not submit, it looks at the queue which the job is in for either the JAMS_PRODUCTION or the JAMS_DEBUG queue characteristics. If it finds one of these characteristics then that characteristic defines if this is a debug or production run. If neither of the characteristics is found on the queue, this field is used to determine if this is a debug or production run.

Submit automatically submitted jobs x before scheduled time

This parameter defines when an automatically submitted job will be submitted to a batch sub-system. At least once a day, the JAMS_AUTOSUBMIT job runs to determine what jobs need to be automatically submitted. These jobs are scheduled by JAMS but they are *not* submitted to the batch sub-system just yet. The scheduled jobs can be viewed and/or managed with the JAMS Job Monitor.

If this parameter is set to zero, the scheduled jobs are submitted to the batch sub-system at their scheduled time. If the parameter is not zero, the jobs are submitted that many minutes before their scheduled time and they hold until their scheduled time.

Keep Completed Jobs in the Monitor for...

Completed jobs will remain in the JAMS Job Monitor display for this length of time after they have completed.

Scan the OpenVMS queues for unknown Jobs every...

The JAMS Schedule process can periodically scan the OpenVMS batch queues for jobs which are not known to JAMS. A job is known to JAMS if it was submitted by JAMS or if the job has started and has run the JAMS_REGISTRAR.EXE program.

Once a job is known to JAMS, it will be displayed on the *Show Current Jobs* and *Monitor Current Jobs* screens.

If you set this field to zero, JAMS will never scan the OpenVMS batch queues. This would be appropriate if all of your jobs are submitted with JAMS or if you do not want to see jobs on the JAMS Monitor screen until they have started execution.

If you have performance problems with the OpenVMS queue sub-system, you may be able to alleviate the problem by setting this field to a fairly small number, such as 5 or 10 minutes, and then encouraging users to use the JAMS functions to display their jobs rather than the OpenVMS SHOW QUEUE command. Since the JAMS functions do not access the OpenVMS queue database, you have one process scanning the OpenVMS queue database every 5 or 10 minutes instead of many processes (people) scanning the queue database every 1 or 2 minutes.

Checking known jobs vs. OpenVMS queues every...

The JAMS Schedule process will periodically check its known jobs against the OpenVMS queues. This is done to resolve any inconsistencies between the JAMS list of known jobs and the OpenVMS queues. Inconsistencies can be introduced for the following reasons:

- 1 JAMS has not been started but jobs are run anyway.
- 2 The JAMS_SCHEDULE process is down and a job is deleted while it is still pending in a batch queue.
- 3 The JAMS_SCHEDULE process is down and a job completes *without* having run the JAMS_REGISTRAR program.

The JAMS Database Configuration

- 4 A pending jobs hold status and/or after time is changed with DCL commands rather than the *JAMS* Job Monitor.

Only the last item in this list should happen under normal conditions. The last item is not critical because the hold status and after time in the *JAMS* known jobs list is only used for reference. If you use the *JAMS* Job Monitor to reschedule the job, the correct attributes are retrieved from the OpenVMS queue database.

This check always occurs when the JAMS_SCHEDULE process starts up. This value determines the interval between checks. You cannot turn this check off. If you set this field to a value which is less than 5 minutes, the check will occur every 5 minutes. We suggest a value of 60 to 120 minutes. A longer value would be appropriate if all of your job rescheduling is performed with the *JAMS* Job Monitor.

Maximum Size of a .LOG file to include in mail message

Defines the maximum size, in disk blocks, of a .LOG file which should be included in a mail message. If the .LOG file is larger than this, the mail message will include the full name of the .LOG file and an indication that it was too large to include.

1.12 Node Definitions

Node Definitions define remote nodes which are also running *JAMS*. *JAMS* Node Definitions are used to validate the node name entered in a remote job dependency definition. Also, a node must be defined before it can be added to a Node Group.

Note: Do not define the nodes which comprise your local VAXcluster. You should only define nodes which are remote and accessed via DECnet.

Node Name

This is the DECnet node name of the remote node which you are defining.

Description

The description should be used to accurately identify this node definition. It is for reference purposes only.

1.13 Node Groups

Node Groups are groups of remote nodes which can be treated as a single node. *JAMS* Node and Node Group Definitions are used to validate the node name entered in a remote job dependency definition. A Node Group is a list of remote nodes which can be referred to by its group name as though it were a node.

In many environments, there is a central VAXcluster with many remote, smaller processing sites. If you wanted to make a Job on the central VAXcluster depend upon the completion of another Job at each of the remote sites, you could add a Job Dependency to the Job on the central VAXcluster for each of the remote sites or create a Node Group which includes all of the remote sites and then create only one Job Dependency which refers to the Node Group.

Group Name

This is the name of the Node Group. You can use this name anywhere *JAMS* expects a remote node name.

Description

The description should be used to accurately identify this Node Group definition. It is for reference purposes only.

Node List

This is a list of the remote Nodes which are included in this Node Group.

1.14 Access Control

Access Control Lists are used to define access restrictions for various capabilities within the *JAMS* system. *JAMS* access control makes use of OpenVMS Access Control Lists, also known as ACLs. Refer to the OpenVMS documentation for more information on Access Control Lists.

An ACL is a list of Access Control Entries (or ACEs). An ACE consists of one or more identifiers and the types of access to be granted to users which match the identifiers.

The identifier field can contain one or more user identifiers separated by a plus sign (+). Identifiers can take any of the following forms:

- [1,1] - Numeric UIC identifier.
- [1,*] - Numeric UIC identifier with wildcards.
- [SYSTEM] - Text UIC identifier.
- PAYROLL_USER - User defined rights identifier.
- BATCH - System defined rights identifier.

These identifiers can be combined, for example, "[100,*]+PAYROLL_USER" will only match users in UIC group 100 who have the PAYROLL_USER rights identifier.

When *JAMS* checks to determine if a user can perform a function, it will start at the top of the ACL and check the identifiers specified in each ACE against the identifiers held by the user. When a match is found, the user is granted only the access specified on the ACE which matched, subsequent ACEs in the list are not checked. If the end of the list is reached without finding a match, no access is granted.

The Access fields on the ACE Maintenance Screen will vary based on the function whose security you are modifying. The Access fields accept a Y (yes) or N (no) entry to either grant or deny the specified access capability. The Access capabilities for each of the security functions are explained in the following sections.

1.14.1 History Inquiry

History Inquiry has only one security option, Inquiry. You can either grant or deny access to the History Inquiry application.

1.14.2 Job Monitor Access

Job Monitor security has the following access types:

The JAMS Database

Access Control

Access	Description
Execute	Allows access to the Job Monitor. Only Jobs which the user has MONITOR access to will be displayed. MONITOR access is defined in the Jobs System definition.
See_All_Jobs	Allows access to the Job Monitor and the ability to monitor jobs submitted by anyone.
See_Own_Jobs	Allows access to the Job Monitor but only jobs submitted by the user running the monitor will be displayed.
Operator	Allows a person to reschedule, hold, release and delete any job which is on their display.
Abort_Jobs	Allows a person to abort and restart any job which is on their display.

Monitor capabilities are also controlled by System definitions. For example, you could grant someone SEE_ALL_JOBS access to the Job Monitor which would let them monitor all batch jobs. Then each System definition could define whether or not the user can manage or abort Jobs which are in the System.

1.14.3 Setup Definitions

Setup Definitions has the following access types:

Access	Description
Add	Allows addition of new Setup definitions
Change	Allows modification of existing Setup definitions
Inquire	Allows inquiry into Setup definitions
Delete	Allows deletion of Setup definitions

Setup definitions are also controlled by the Access Control List of each System definition. To create a Setup, you need to have ADD access to Setup definitions and you must have SUBMIT access to the System to which the Setup's Job definition belongs and, DEFINE_SETUP access to the System to which the Setup definition belongs.

The SUBMIT and DEFINE_SETUP access rights are defined for each System definition. Refer to the Section 1.2, System Definitions for more information on Systems and their security.

1.14.4 Job Definitions

The Job Definitions function has the following access types:

Access	Description
Add	Allows addition of new Job definitions
Change	Allows modification of existing Job definitions
Inquire	Allows inquiry into Job definitions
Delete	Allows deletion of Job definitions

Job definitions are also controlled by the Access Control List of each System definition. To create a Job, you need to have ADD access to Job definitions, and you must have JOB_ADD access to the System to which the Job belongs. Similarly, to modify, delete or inquire into a Job definition, you must have the corresponding JOB_CHANGE, JOB_DELETE or JOB_INQUIRE access right for the System to which the Job belongs.

Refer to the Section 1.2, System Definitions for more information on Systems and their security.

1.14.5 System Definitions

System Definitions has the following access types:

Access	Description
Control	Allows modification of a Systems individual ACL
Add	Allows addition of new System definitions
Change	Allows modification of existing System definitions
Inquire	Allows inquiry into System definitions
Delete	Allows deletion of System definitions

Each System Definition has it's own access control information. This ACL can be viewed and/or modified from the System Definitions menu option.

Note that in order to modify, delete or view a System definition, you must have CHANGE, DELETE or INQUIRE access to System definitions and CHANGE, DELETE or INQUIRE access to the specific System definition which you want to manipulate.

1.14.6 Menu Definitions

Menu Definitions has the following access types:

Access	Description
Add	Allows addition of new Menu definitions
Change	Allows modification of existing Menu definitions
Inquire	Allows inquiry into Menu definitions
Delete	Allows deletion of Menu definitions

1.14.7 Variable Definitions

The Variable Definitions function has the following access types:

Access	Description
Control	Allows modification of Variables individual ACLs, provided that the Variables individual ACL also grants CONTROL.
Add	Allows addition of new Variable definitions
Change	Allows modification of existing Variable definitions
Inquire	Allows inquiry into Variable definitions
Delete	Allows deletion of Variable definitions

Each Variable has an individual ACL which is used to protect only that Variable.

1.14.8 Trigger Definitions

The Trigger Definitions function has the following access types:

Access	Description
Reset	Allows use of the RESET command.
Manage	Allows use of the ENABLE and DISABLE commands.
Add	Allows addition of new Trigger definitions
Change	Allows modification of existing Trigger definitions
Inquire	Allows inquiry into Trigger definitions
Delete	Allows deletion of Trigger definitions

1.14.9 Date Definitions

Date Maintenance has the following access types:

Access	Description
Add	Allows addition of new Date definitions
Change	Allows modification of existing Date definitions
Inquire	Allows inquiry into Date definitions
Delete	Allows deletion of Date definitions

1.14.10 Configuration

Configuration has only one security option, Execute. You can either grant or deny access to the Configuration application.

1.14.11 Date Type Definitions

Date Type Maintenance has the following access types:

Access	Description
Add	Allows addition of new Date Types
Change	Allows modification of existing Date Types
Inquire	Allows inquiry into Date Types
Delete	Allows deletion of Date Types

1.14.12 Access Control

Access Control has two security options, Execute and Inquiry. Execute access grants the ability to manipulate the Access Control Lists for all security options.

1.14.13 Node Definitions

The Node Maintenance security option controls the ability to access Node and Node Group definitions. Node Maintenance has the following access types:

Access	Description
Add	Allows addition of new Node definitions
Change	Allows modification of existing Node definitions
Inquire	Allows inquiry into Node definitions
Delete	Allows deletion of Node definitions

1.14.14 Named Times Access

The Named Times Access security option controls the ability to access Named Time Definitions. Named Time Access has the following access types:

Access	Description
Manage	Allows access to the ENABLE TIME and DISABLE TIME commands
Add	Allows addition of new Named Time definitions
Change	Allows modification of existing Named Time definitions
Inquire	Allows inquiry into Named Time definitions
Delete	Allows deletion of Named Time definitions

1.14.15 Username Access

The Username Access security option controls the ability to use the SET USERNAME and EXTRACT USERNAME commands. Username Access has the following access types:

Access	Description
Execute	Allows use of the SET USERNAME command
Inquire	Allows use of the EXTRACT USERNAME

1.14.16 Resource Access

The Resource Access security option controls the ability to access Resource Definitions. Resource Access has the following access types:

Access	Description
Add	Allows addition of new Resource definitions using the CREATE RESOURCE/NOREPLACE command
Change	Allows modification of existing Resource definitions using the CREATE RESOURCE/REPLACE or the SET RESOURCE commands
Inquire	Allows inquiry into existing Resource definitions using the EXTRACT RESOURCE or SHOW RESOURCE commands
Delete	Allows deletion of Resource definitions using the DELETE RESOURCE command

2 JAMS Commands

This chapter describes the *JAMS* commands.

2.1 Command Environment

The following sections explain the commands available from within the *JAMS_MASTER* program.

ACQUIRE RESOURCE

ACQUIRE RESOURCE

Acquire a quantity of a *JAMS* Resource.

FORMAT	ACQUIRE RESOURCE <i>resource-name quantity</i>
---------------	---

PARAMETERS	<i>resource-name</i> A valid <i>JAMS</i> Resource name.
	<i>quantity</i> The amount of the resource required.

DESCRIPTION	The ACQUIRE RESOURCE command is used to acquire a quantity of a specific <i>JAMS</i> Resource.
--------------------	--

restrictions	You must have ACQUIRE access to the Resource whose value you are trying to obtain.
---------------------	--

related commands	• CREATE RESOURCE • DELETE RESOURCE • RELEASE RESOURCE • SET RESOURCE • SHOW RESOURCE
-------------------------	---

QUALIFIERS	<i>/LOG</i> Specifies that the Resource acquisition activity should be logged.
	<i>/TIME_OUT=seconds</i> Specifies the number of seconds to wait to for the resource to become available. The default is to wait forever. If the specified quantity of the Resource is not available within the time out period, the command fails.

CREATE DATE

Create or replace a *JAMS* Date Definition.

FORMAT **CREATE DATE** *date-specification*

PARAMETERS ***date-specification***

A valid *JAMS* date specification. This specification can be an English language date specification which will be evaluated into an absolute date.

DESCRIPTION The CREATE DATE command is used to create or replace a *JAMS* Date Definition.

restrictions You must have ADD access to Date Definitions.

related commands

- CREATE DTYPE
- EXTRACT DATE
- EXTRACT DTYPE

QUALIFIERS ***/DESCRIPTION***

This qualifier can be used to specify a description of the Date.

/REPLACE

/NOREPLACE

This qualifier is used to specify whether you want to create or replace the Definition. If you omit this qualifier, the Definition will be created if it does not exist and replaced if it does exist. If you specify */REPLACE*, an error will be returned if the Definition does not exist and no changes will be made. If you specify */NOREPLACE*, an error will be returned if the Definition exists and no changes will be made.

/SPECIFIC_TYPE=specific-type

Specifies the Specific Date Type of this Date Definition.

/TODAY_IS=date

The date specification will be interpreted as if today was the date specified with this qualifier. If you do not use this qualifier, the default is the current system date.

/TYPE=date-type

Specifies the Date Type of this Date Definition. This qualifier is required.

CREATE DATE

/WORKDAY ***/NOWORKDAY***

Specifies whether this Date Specification changes the workday status of the date. Omitting this qualifier signifies that this Date Definition does not change the workday status of the date. Specifying */WORKDAY* will make this date a workday and */NOWORKDAY* will make it a non-workday.

EXAMPLES

```
JAMS> CREATE DATE "3RD THURSDAY OF NOVEMBER" -  
_JAMS> /TYPE=HOLIDAY/SPECIFIC=THANKSGIVING/NOWORKDAY -
```

In this example, Thanksgiving is added as a HOLIDAY Date Type. Note that this CREATE DATE command does not need to be changed from year to year.

```
JAMS> CREATE DATE 5-AUG-1996/TYPE=FISCAL
```

In this example, August 5, 1996 is added as the start date of Date Type FISCAL.

CREATE DTYPE

Create or replace a *JAMS* Date Type.

FORMAT **CREATE DTYPE** *date-type-name*

PARAMETERS ***date-type-name***
A valid *JAMS* Date Type name.

DESCRIPTION The CREATE DTYPE command is used to create or replace a Date Type Definition.

restrictions You must have ADD access to Date Type Definitions.

related commands

- CREATE DATE
- EXTRACT DTYPE

QUALIFIERS ***/CONTINUOUS***
/NOCONTINUOUS (default)
 Used to specify whether or not this is a continuous Date Type.

/DESCRIPTION=description
 This qualifier can be used to specify a description of the new Date Type.

/REPLACE
/NOREPLACE
 This qualifier is used to specify whether you want to create or replace the Definition. If you omit this qualifier, the Definition will be created if it does not exist and replaced if it does exist. If you specify */REPLACE*, an error will be returned if the Definition does not exist and no changes will be made. If you specify */NOREPLACE*, an error will be returned if the Definition exists and no changes will be made.

/SPECIFIC_TYPES=specific-types
 This qualifier is used to specify a list of valid Specific Date Types for this Date Type. The list should be separated by commas and enclosed in parentheses.

CREATE METHOD

CREATE METHOD

Create or replace a *JAMS* Execution Method.

FORMAT	CREATE METHOD <i>method-name</i>
---------------	---

PARAMETERS	<i>method-name</i> A valid <i>JAMS</i> Method name.
-------------------	---

DESCRIPTION	The CREATE METHOD command is used to create or replace a <i>JAMS</i> Execution Method Definition.
--------------------	---

restrictions	You must have execution access to <i>JAMS</i> Configuration settings.
---------------------	---

related commands	• DELETE METHOD • EXTRACT METHOD
-------------------------	--

QUALIFIERS	<i>/COMMAND=command</i> This qualifier is used to specify a command which is sent to a <i>JAMS</i> Agent to execute the job's command procedure. If the command contains "%s", the name of the temporary command procedure is inserted in it's place. <i>/DESCRIPTION=description</i> This qualifier can be used to specify a description of the new Method. <i>/EXTENSION=file-extension</i> This qualifier is used to specify the default file extension for a job's command procedure. <i>/JOB_MODULE=module-name</i> The <i>/JOB_MODULE</i> qualifier can be used to specify the name of a text module in the template library which is used as the starting point for parsing the job's command procedure. Specifying a value with this qualifier implies that the job is parsed. <i>/OPTIONS=option-list</i> The <i>/OPTIONS</i> qualifier can be used to specify one or more of the following options: <table><tr><td>COMMAND</td><td>Specifies that this method uses a command instead of a command procedure. The job's source is not a file specification but is instead a single command.</td></tr></table>	COMMAND	Specifies that this method uses a command instead of a command procedure. The job's source is not a file specification but is instead a single command.
COMMAND	Specifies that this method uses a command instead of a command procedure. The job's source is not a file specification but is instead a single command.		

INTERACTIVE	Specifies that this method will run a Windows NT job interactively. When running, the job will have access to the default Window Station and display.
PARSE	Specifies that the procedure specified with the /PROCEDURE qualifier is the name of a text module in the template library. This specifies that the procedure should be parsed, possibly in addition to parsing the job's source.

/PROCEDURE=procedure-name

The /PROCEDURE qualifier can be used to specify the name of a command procedure or text module in the template library which is used as the starting point for parsing and building a temporary command procedure. This procedure is executed by JAMS to indirectly execute a job.

/REPLACE

/NOREPLACE

This qualifier is used to specify whether you want to create or replace the Definition. If you omit this qualifier, the Definition will be created if it does not exist and replaced if it does exist. If you specify /REPLACE, an error will be returned if the Definition does not exist and no changes will be made. If you specify /NOREPLACE, an error will be returned if the Definition exists and no changes will be made.

/TYPE=execution-type

Specifies what type of execution method this is. The supported types are:

AGENT	Runs the job via a JAMS Agent. The procedure specified with the /PROCEDURE qualifier is executed in a detached process to send the job's command procedure to the Agent.
DETACH	Executes the job's command procedure as a detached process.
PROCEDURE	Runs the job indirectly via the command procedure specified with the /PROCEDURE qualifier. This procedure is executed as a detached process.
SUBMIT	Submits the job's command procedure to an OpenVMS batch queue.

CREATE RESOURCE

CREATE RESOURCE

Create or replace a *JAMS* Resource.

FORMAT	CREATE RESOURCE <i>resource-name</i>
---------------	---

PARAMETERS	<i>resource-name</i> A valid <i>JAMS</i> Resource name.
-------------------	---

DESCRIPTION	The CREATE RESOURCE command is used to create or replace a <i>JAMS</i> Resource Definition.
--------------------	---

restrictions	You must have execution access to <i>JAMS</i> Configuration settings.
---------------------	---

related commands	• ACQUIRE RESOURCE • DELETE RESOURCE • EXTRACT RESOURCE • RELEASE RESOURCE • SET RESOURCE • SHOW RESOURCE
-------------------------	---

QUALIFIERS	<i>/ACL=access-control-list</i> This qualifier is used to specify an access control list for the Resource.
-------------------	--

	<i>/AVAILABLE=quantity-available</i> This qualifier is used to specify the quantity of this resource that is available.
--	---

	<i>/BY_NODE</i>
--	------------------------

	<i>/NOBY_NODE (default)</i> This qualifier can be used to specify that this resource is measured separately for each node. If <i>/BY_NODE</i> is specified, the quantity specified with the <i>/AVAILABLE</i> qualifier is available on each node being managed by <i>JAMS</i> . To specify a different available quantity for a node, use the SET RESOURCE command.
--	--

	<i>/DESCRIPTION=description</i> This qualifier can be used to specify a description of the new Resource.
--	--

/REPLACE***/NOREPLACE***

This qualifier is used to specify whether you want to create or replace the Definition. If you omit this qualifier, the Definition will be created if it does not exist and replaced if it does exist. If you specify */REPLACE*, an error will be returned if the Definition does not exist and no changes will be made. If you specify */NOREPLACE*, an error will be returned if the Definition exists and no changes will be made.

CREATE TIME

CREATE TIME

Create or replace a *JAMS* Named Time.

FORMAT	CREATE TIME <i>time-name</i>
---------------	-------------------------------------

PARAMETERS	<i>time-name</i> A valid <i>JAMS</i> Time name.
-------------------	---

DESCRIPTION	The CREATE TIME command is used to create or replace a <i>JAMS</i> Named Time Definition.
--------------------	---

restrictions	You must have ADD access to Named Time Definitions.
---------------------	---

related commands	• DISABLE TIME • ENABLE TIME • EXTRACT TIME
-------------------------	---

QUALIFIERS	<i>/DESCRIPTION=description</i> This qualifier can be used to specify a description of the new Variable.
-------------------	--

/DISABLE

/NODISABLE (default)

Specifies whether or not this Time is automatically disabled at it's end time.

/ENABLE

/NOENABLE (default)

Specifies whether or not this Time is automatically enabled at it's start time.

/END=end-time

Specifies the Times end time.

/REPLACE

/NOREPLACE

This qualifier is used to specify whether you want to create or replace the Definition. If you omit this qualifier, the Definition will be created if it does not exist and replaced if it does exist. If you specify */REPLACE*, an error will be returned if the Definition does not exist and no changes will be made. If you specify */NOREPLACE*, an error will be returned if the Definition exists and no changes will be made.

/START=start-time

Specifies the Times start time.

CREATE VARIABLE

CREATE VARIABLE

Creates a new *JAMS* Variable.

FORMAT	CREATE VARIABLE <i>variable-name</i> [<i>value</i>]
---------------	--

PARAMETERS	<i>variable-name</i> A valid <i>JAMS</i> Variable name.
-------------------	---

value

The value for the specified Variable. Enclose this in quotes to preserve case and embedded spaces.

If the Variable has a datatype of DATE, the new value must be specified in DD-MMM-YYYY format. If the Variable has a datatype of TIME, the new value must be specified in HH:MM:SS, 24 hour format.

DESCRIPTION	The CREATE VARIABLE command is used to create a new <i>JAMS</i> Variable.
--------------------	---

restrictions	You must have ADD access to Variable Definitions.
---------------------	---

related commands	• DELETE VARIABLE • GET VARIABLE • SET VARIABLE • SHOW VARIABLE
-------------------------	---

QUALIFIERS	<i>/ACL</i> This qualifier is used to specify an Access Control List for the Variable.
-------------------	--

/DATATYPE

This qualifier is used to specify the data type of the new Variable. If you do not specify a data type, the default is TEXT. The valid data types are:

TEXT
INTEGER
BOOLEAN
DATE
TIME
DATETIME
FLOAT
UNKNOWN

/DESCRIPTION

This qualifier can be used to specify a description of the new Variable.

/REPLACE

/NOREPLACE

This qualifier is used to specify whether you want to create or replace the Variable. If you omit this qualifier, the Variable will be created if it does not exist and replaced if it does exist. If you specify /REPLACE, an error will be returned if the Variable does not exist and no changes will be made. If you specify /NOREPLACE, an error will be returned if the Variable exists and no changes will be made.

DEFINE JOB

DEFINE JOB

Creates or replaces a Job Definition.

FORMAT **DEFINE JOB** *job-name*

PARAMETERS ***job-name***
The Job Name of the Job which you want to create or replace.

DESCRIPTION The DEFINE JOB command is used to create or replace an entire Job Definition. The DEFINE JOB command marks the beginning of a Job definition. The JAMS> prompt changes to JOB_DEF> until the Job Definition is complete.

Usually, the DEFINE JOB command is part of a *JAMS* command procedure which contains the entire definition. You can use the EXTRACT JOB command to create a complete DEFINE JOB command procedure.

restrictions You must be authorized to create or replace the Job Definition. Additional security checks are made based upon the contents of the Job Definition.

related commands

- EXTRACT JOB

QUALIFIERS ***/REPLACE***

This qualifier is used to specify whether you want to create or replace the Definition. If you omit this qualifier, the Definition will be created if it does not exist and replaced if it does exist. If you specify /REPLACE, an error will be returned if the definition does not exist and no changes will be made. If you specify /NOREPLACE, an error will be returned if the definition exists and no changes will be made.

SYNTAX The complete syntax for a Job definition is:
DIAGRAM

```
DEFINE JOB job-name
    SYSTEM system-id
    [ DESCRIPTION quoted-string [, quoted-string] ]
    [ BATCH QUEUE queue-name ]
    [ PRIORITY MODIFIER integer ]
    [ USERNAME username ]
    [ RETAIN { ALWAYS
              ERROR
              FOR delta-time
              NEVER } ]
```

```

[ SUBMIT ON HOLD ]
[ RESTARTABLE ]
[ SUPPRESS MENU ]
[ NOTIFY SUBMITTER ]
[ [NO] KEEP LOGS ]
[ [NO] PRINT LOGS ]
[ [NO] TIMESTAMP LOGS ]
[ SCHEDULE WINDOW named-time ]
[ SCHEDULE RANGE time TO time ]
[
  MISSED ACTION {
    ABORT
    CONTINUE
    NONE
    RESCHEDULE
    RESTART
  } ]
[ NOTIFY MISSED ]
[ DEFAULT SUBMIT DATE date-specification ]
[ DEFAULT SUBMIT TIME time ]
[ SCHEDULED DATE date-specification ]
[ EXCEPT FOR date-specification ]
[ SCHEDULED TIME time ]
[
  RESUBMIT EVERY delta-time UNTIL end-time
  [
    BASED ON {
      SCHEDULED
      START
      END
    } TIME
  ]
]
[ [NO] AUTO SUBMIT ]
[
  NONWORKDAYS {
    IGNORE
    DEFER
    SCHEDULE
  } ]
[ parameter-specification... ]
[ report-specification... ]
[ dependency-specification... ]
[
  { SOURCE FILE file-specification }
  { COMMAND command-line }
]
[ METHOD method-name ]
[ AGENT NODE agent-node-name ]
[ TIMESTAMP LOGS ]
[ NO TIMESTAMP LOGS ]
[ resource-specification... ]
[
  SOURCE IS
  jobs source code
  END_SOURCE
]
[ PRECHECK JOB job-name ]
[ PRECHECK INTERVAL delta-time ]
[
  RECOVERY SEVERITY {
    SUCCESS
    INFORMATIONAL
    WARNING
    ERROR
    FATAL
  } ]
[ RECOVERY JOB job-name ]
[
  RECOVERY INSTRUCTIONS
  recovery instructions
  END_RECOVERY
]
[ MAIL NOTIFICATION quoted-string ]

```

DEFINE JOB

```
[ REPLY NOTIFICATION quoted-string ]
[ OPERATOR NOTIFICATION quoted-string ]
[ NOTIFICATION JOB job-name ]
[
  [
    NOTIFY SEVERITY {
      SUCCESS
      INFORMATIONAL
      WARNING
      ERROR
      FATAL
      NEVER
    }
  ]
]
[ STALLED TIME delta-time ]
[ RUNAWAY ELAPSED TIME delta-time ]
[ RUNAWAY ELAPSED PERCENT integer ]
[ RUNAWAY CPU TIME delta-time ]
[ RUNAWAY CPU PERCENT integer ]
[ SHORT ELAPSED TIME delta-time ]
[ SHORT ELAPSED PERCENT integer ]
[ SHORT CPU TIME delta-time ]
[ SHORT CPU PERCENT integer ]
[
  [
    SHORT SEVERITY {
      SUCCESS
      INFORMATIONAL
      WARNING
      ERROR
      FATAL
    }
  ]
]
[ ACL (acl) ]
END_JOB
```

The syntax for a parameter specification is:

```
PARAMETER parameter-name
[ PROMPT quoted-string ]
[
  {
    TEXT LENGTH integer
    INTEGER LENGTH integer
    DATE
    TIME
  }
]
[ DEFAULT VARIABLE variable-name ]
[ DEFAULT FORMAT format ]
[ DEFAULT VALUE quoted-string ]
[ [NO] REQUIRED ]
[ [NO] MUST_FILL ]
[ [NO] ALLOW_ENTRY ]
[ [NO] UPPERCASE ]
[ [NO] HIDE ]
[ HELP quoted-string ]
END_PARAMETER
```

The syntax for a report specification is:

```
REPORT report-id
[ DESCRIPTION quoted-string ]
[ LOGICAL logical-name ]
[ QUEUE queue-name ]
[ FORM form-name ]
[ COPIES integer ]
```

```
[ RETENTION integer ]
[ FILENAME file-specification ]
[ QUALIFIERS quoted-string ]
END_REPORT
```

The syntax for a job dependency is:

```
DEPENDS ON JOB [node::]job-name
[
  SEVERITY {
    SUCCESS
    INFORMATIONAL
    WARNING
    ERROR
    FATAL
  }
]
{
  SINCE JOB job-name
  [
    SEVERITY {
      SUCCESS
      INFORMATIONAL
      WARNING
      ERROR
      FATAL
    }
  ]
}
[ [NO] SINCE JOB
  [ WITHIN [integer DAYS [AND]] [integer HOURS] ]
  [ WAIT IF QUEUED ]
]
END_DEPEND
```

The syntax for a variable dependency is:

```
DEPENDS ON VARIABLE variable-id
{
  CHANGED
  TRUE
  FALSE
  = value
  < value
  > value
  <= value
  >= value
  <> value
}
END_DEPEND
```

DEFINE MENU

Creates or replaces a Menu Definition.

FORMAT	DEFINE MENU <i>menu-name</i>
---------------	-------------------------------------

PARAMETERS	<i>menu-name</i> The Menu Name of the Menu which you want to create or replace.
-------------------	---

DESCRIPTION	The DEFINE MENU command is used to create or replace an entire Menu Definition. The DEFINE MENU command marks the beginning of a Menu definition. The JAMS> prompt changes to MENU_DEF> until the Menu Definition is complete.
--------------------	--

Usually, the DEFINE MENU command is part of a *JAMS* command procedure which contains the entire definition. You can use the EXTRACT MENU command to create a complete DEFINE MENU command procedure.

restrictions	You must be authorized to create or replace Menu Definitions.
---------------------	---

related commands	• EXTRACT MENU
-------------------------	--

QUALIFIERS	<i>/REPLACE</i> This qualifier is used to specify whether you want to create or replace the Definition. If you omit this qualifier, the Definition will be created if it does not exist and replaced if it does exist. If you specify /REPLACE, an error will be returned if the definition does not exist and no changes will be made. If you specify /NOREPLACE, an error will be returned if the definition exists and no changes will be made.
-------------------	--

SYNTAX	The complete syntax for a Menu definition is:
---------------	---

DIAGRAM

```
DEFINE MENU menu-name
    [ DESCRIPTION quoted-string ]
    [ selection-specification... ]
END_MENU
```

The syntax for a selection specification is:

```
SELECTION select-name-specification
INCLUDE ([JOBS,] [SETUPS,] [MENUS])
[ SYSTEM system-id ]
```

DEFINE SETUP

Creates or replaces a Setup Definition.

FORMAT **DEFINE SETUP** *setup-name*

PARAMETERS ***setup-name***
The Setup Name of the Setup which you want to create or replace.

DESCRIPTION The DEFINE SETUP command is used to create or replace an entire Setup Definition. The DEFINE SETUP command marks the beginning of a Setup Definition. The JAMS> prompt changes to SETUP_DEF> until the Setup Definition is complete.

Usually, the DEFINE SETUP command is part of a *JAMS* command procedure which contains the entire definition. You can use the EXTRACT SETUP command to create a complete DEFINE SETUP command procedure.

restrictions You must be authorized to create or replace the Setup Definition. Additional security checks are made based upon the contents of the Setup Definition.

related commands

- EXTRACT SETUP

QUALIFIERS ***/REPLACE***

This qualifier is used to specify whether you want to create or replace the Definition. If you omit this qualifier, the Definition will be created if it does not exist and replaced if it does exist. If you specify */REPLACE*, an error will be returned if the definition does not exist and no changes will be made. If you specify */NOREPLACE*, an error will be returned if the definition exists and no changes will be made.

SYNTAX The complete syntax for a Setup definition is:

DIAGRAM

```

DEFINE SETUP setup-name
    [ DESCRIPTION quoted-string [, quoted-string] ]
    [ SYSTEM system-id ]
    [ PRIORITY MODIFIER integer ]
    [ [NO] SUBMIT ON HOLD ]
    [ SUPPRESS MENU ]
    [ NOTIFY SUBMITTER ]

```

DEFINE SETUP

```

[ SCHEDULE WINDOW named-time ]
[ SCHEDULE RANGE time TO time ]
[ MISSED ACTION { ABORT
                  CONTINUE
                  NONE
                  RESCHEDULE
                  RESTART } ]
[ NOTIFY MISSED ]
[ DEFAULT SUBMIT DATE date-specification ]
[ DEFAULT SUBMIT TIME time ]
[ SCHEDULED DATE date-specification ]
[ EXCEPT FOR date-specification ]
[ SCHEDULED TIME time ]
[ RESUBMIT EVERY delta-time UNTIL end-time ]
[ [ BASED ON { SCHEDULED
               START
               END } TIME ] ]
[ [NO] AUTO SUBMIT ]
[ NONWORKDAYS { IGNORE
               DEFER
               SCHEDULE } ]
[ dependency-specification... ]
[ PRECHECK JOB job-name ]
[ PRECHECK INTERVAL delta-time ]
[ RETAIN { ALWAYS
           NEVER
           ERROR
           FOR delta-time } ]
[ RECOVERY SEVERITY { SUCCESS
                     INFORMATIONAL
                     WARNING
                     ERROR
                     FATAL } ]
[ RECOVERY JOB job-name ]
[ RECOVERY INSTRUCTIONS ]
[ recovery instructions ]
[ END_RECOVERY ]
[ MAIL NOTIFICATION quoted-string ]
[ REPLY NOTIFICATION quoted-string ]
[ OPERATOR NOTIFICATION quoted-string ]
[ NOTIFICATION JOB job-name ]
[ NOTIFY SEVERITY { SUCCESS
                   INFORMATIONAL
                   WARNING
                   ERROR
                   FATAL
                   NEVER } ]
[ LOG FILE file-specification ]
[ [NO] KEEP LOGS ]
[ [NO] PRINT LOGS ]
[ [NO] TIMESTAMP LOGS ]
[ STALLED TIME delta-time ]
[ RUNAWAY ELAPSED TIME delta-time ]

```



```
[ RUNAWAY ELAPSED PERCENT integer ]
[ SHORT ELAPSED TIME delta-time ]
[ SHORT ELAPSED PERCENT integer ]
[
  SHORT SEVERITY {
    SUCCESS
    INFORMATIONAL
    WARNING
    ERROR
    FATAL
  }
]
[ ACL (acl) ]
[ resource-requirement... ]
[ job-specification... ]
END_SETUP
```

The syntax for a job specification is:

JOB job-name

```
[ NAME override-job-name ]
[ DESCRIPTION quoted-string [, quoted-string] ]
[ STEP step-number ]
[ [NO] WAIT ]
[ CONTINUE ]
[
  MINIMUM SEVERITY {
    SUCCESS
    INFORMATIONAL
    WARNING
    ERROR
    FATAL
  }
]
[ AGENT NODE agent-name ]
[ BATCH QUEUE queue-name ]
[ PRIORITY MODIFIER integer ]
[ USERNAME username ]
[ [NO] SUBMIT ON HOLD ]
[ SCHEDULE WINDOW named-time ]
[ SCHEDULE RANGE time TO time ]
[
  MISSED ACTION {
    ABORT
    CONTINUE
    NONE
    RESCHEDULE
    RESTART
  }
]
[ NOTIFY MISSED ]
[ SCHEDULED DATE date-specification ]
[ EXCEPT FOR date-specification ]
[
  NONWORKDAYS {
    IGNORE
    SCHEDULE
  }
]
[ parameter-specification... ]
[ report-specification... ]
[ PRECHECK JOB job-name ]
[ PRECHECK INTERVAL delta-time ]
[
  RECOVERY SEVERITY {
    SUCCESS
    INFORMATIONAL
    WARNING
    ERROR
    FATAL
  }
]
[ RECOVERY JOB job-name ]
```

DEFINE SETUP

```
[ MAIL NOTIFICATION quoted-string ]
[ REPLY NOTIFICATION quoted-string ]
[ OPERATOR NOTIFICATION quoted-string ]
[ NOTIFICATION JOB job-name ]
[
    [
        NOTIFY SEVERITY {
            SUCCESS
            INFORMATIONAL
            WARNING
            ERROR
            FATAL
            NEVER
        }
    ]
]
[ STALLED TIME delta-time ]
[ RUNAWAY ELAPSED TIME delta-time ]
[ RUNAWAY ELAPSED PERCENT integer ]
[ RUNAWAY CPU TIME delta-time ]
[ RUNAWAY CPU PERCENT integer ]
[ SHORT ELAPSED TIME delta-time ]
[ SHORT ELAPSED PERCENT integer ]
[ SHORT CPU TIME delta-time ]
[ SHORT CPU PERCENT integer ]
[
    [
        SHORT SEVERITY {
            SUCCESS
            INFORMATIONAL
            WARNING
            ERROR
            FATAL
        }
    ]
]
END_JOB
```

The syntax for a parameter specification is:

```
PARAMETER parameter-name
[ PROMPT quoted-string ]
[ GLOBAL parameter-name ]
[ DEFAULT VARIABLE variable-name ]
[ DEFAULT VALUE quoted-string ]
[ [NO] REQUIRED ]
[ [NO] MUST_FILL ]
[ [NO] ALLOW_ENTRY ]
[ [NO] UPPERCASE ]
[ [NO] HIDE ]
END_PARAMETER
```

The syntax for a report specification is:

```
REPORT report-id
[ QUEUE queue-name ]
[ FORM form-name ]
[ COPIES integer ]
[ RETENTION integer ]
[ QUALIFIERS quoted-string ]
END_REPORT
```

The syntax for a job dependency is:

```
DEPENDS ON JOB [node::]job-name
[
  SEVERITY {
    SUCCESS
    INFORMATIONAL
    WARNING
    ERROR
    FATAL
  }
]
{
  SINCE JOB job-name
  [
    SEVERITY {
      SUCCESS
      INFORMATIONAL
      WARNING
      ERROR
      FATAL
    }
  ]
  [NO] SINCE JOB
  [ WITHIN [integer DAYS [AND]] [integer HOURS] ]
  [ WAIT IF QUEUED ]
}
END_DEPEND
```

The syntax for a variable dependency is:

```
DEPENDS ON VARIABLE variable-id
{
  CHANGED
  TRUE
  FALSE
  = value
  < value
  > value
  <= value
  >= value
  <> value
}
END_DEPEND
```

DEFINE SYSTEM

DEFINE SYSTEM

Creates or replaces a System Definition.

FORMAT	DEFINE SYSTEM <i>system-id</i>
---------------	---------------------------------------

PARAMETERS	<i>system-id</i> The System I.D. of the System which you want to create or replace.
-------------------	---

DESCRIPTION	The DEFINE SYSTEM command is used to create or replace an entire System Definition. The DEFINE SYSTEM command marks the beginning of a System Definition. The JAMS> prompt changes to SYSTEM_DEF> until the System Definition is complete. Usually, the DEFINE SYSTEM command is part of a JAMS command procedure which contains the entire definition. You can use the EXTRACT SYSTEM command to create a complete DEFINE SYSTEM command procedure.
--------------------	--

restrictions	You must be authorized to create or replace the System Definition.
---------------------	--

related commands	• EXTRACT SYSTEM
-------------------------	--

QUALIFIERS	<i>/REPLACE</i> This qualifier is used to specify whether you want to create or replace the definition. If you omit this qualifier, the definition will be created if it does not exist and replaced if it does exist. If you specify /REPLACE, an error will be returned if the definition does not exist and no changes will be made. If you specify /NOREPLACE, an error will be returned if the definition exists and no changes will be made.
-------------------	--

SYNTAX DIAGRAM	The complete syntax for a System definition is:
---------------------------	---

```
DEFINE SYSTEM system-id
[ DESCRIPTION quoted-string ]
[ ACL (acl) ]
[ MAIL NOTIFICATION quoted-string ]
[ REPLY NOTIFICATION quoted-string ]
[ OPERATOR NOTIFICATION quoted-string ]
[ NOTIFICATION JOB job-name ]
[ STALLED TIME delta-time ]
[ RUNAWAY ELAPSED PERCENT integer ]
[ RUNAWAY CPU PERCENT integer ]
```

```

[ SHORT ELAPSED PERCENT integer ]
[ SHORT CPU PERCENT integer ]
[
  SHORT SEVERITY {
    SUCCESS
    INFORMATIONAL
    WARNING
    ERROR
    FATAL
  }
]
[ BATCH QUEUE queue ]
[
  RETAIN {
    ALWAYS
    NEVER
    ERROR
    FOR delta-time
  }
]
[ PRIORITY MODIFIER integer ]
[ USERNAME username ]
[ [NO] KEEP LOGS ]
[ [NO] PRINT LOGS ]
[ LOG LOCATION directory-specification ]
[ TIMESTAMP LOGS ]
[ LOG QUEUE queue ]
[ PRINT QUEUE queue ]
[ REPORT LOCATION directory-specification ]
[
  {
    REQUIRED
    DEFAULT
  } SOURCE LOCATION directory-specification
]
[ [NO] SECURE LOGICALS ]
[ TEMPLATE LIBRARY file-specification ]
[ DOCUMENTATION LOCATION directory-specification ]
[
  RECOVERY INSTRUCTIONS
  recovery instructions
  END_RECOVERY
]
END_SYSTEM

```

DEFINE TRIGGER

DEFINE TRIGGER

Create or replace a Trigger Definition.

FORMAT	DEFINE TRIGGER <i>trigger-name</i>
---------------	---

PARAMETERS	<i>trigger-name</i> The Name of the Trigger which you want to create or replace.
-------------------	--

DESCRIPTION	The DEFINE TRIGGER command is used to create or replace an entire Trigger Definition. The DEFINE TRIGGER command marks the beginning of a Trigger Definition. The JAMS> prompt changes to TRIGGER_DEF> until the Trigger Definition is complete.
--------------------	--

Usually, the DEFINE TRIGGER command is part of a *JAMS* command procedure which contains the entire definition. You can use the EXTRACT TRIGGER command to create a complete DEFINE TRIGGER command procedure.

restrictions	You must be authorized to create or replace Trigger Definitions. Additional security checks are made based upon the contents of the Trigger Definition.
---------------------	---

related commands	• EXTRACT TRIGGER • SHOW TRIGGER
-------------------------	--

QUALIFIERS	<i>/REPLACE</i> This qualifier is used to specify whether you want to create or replace the Definition. If you omit this qualifier, the Definition will be created if it does not exist and replaced if it does exist. If you specify /REPLACE, an error will be returned if the definition does not exist and no changes will be made. If you specify /NOREPLACE, an error will be returned if the definition exists and no changes will be made.
-------------------	--

SYNTAX	The complete syntax for a Trigger definition is:
---------------	--

DIAGRAM	<pre>DEFINE TRIGGER trigger-name [DESCRIPTION quoted-string] [AUTO RESET] [event-specification...] [action-specification...] END_TRIGGER</pre>
----------------	--

The syntax for an event specification is:

```
EVENT
{ job-event
  variable-event }
END_EVENT
```

Where job-event is:

```
JOB job-name SEVERITY { SUCCESS
                       INFORMATIONAL
                       WARNING
                       ERROR
                       FATAL } OR { BETTER
                                   WORSE }
```

and variable-event is:

```
VARIABLE variable-name { CHANGED
                          TRUE
                          FALSE
                          = value
                          < value
                          > value
                          <= value
                          >= value
                          <> value }
```

The syntax for an action specification is:

```
ACTION SUBMIT JOB job-name
[ DATE date-specification ]
[ TIME time-specification ]
[ HOLD ]
END_ACTION
```

DELETE ENTRY

DELETE ENTRY

Deletes one or more *JAMS* Entries from the current schedule.

FORMAT	DELETE ENTRY <i>job-name</i> <i>JAMS-Entry</i>
---------------	---

PARAMETERS	<i>job-name</i> A <i>JAMS</i> Entry number or the name of an entry in the current schedule. The name may contain wildcards.
-------------------	--

DESCRIPTION	The DELETE ENTRY command is used to delete one or more <i>JAMS</i> Entries from the current schedule. If the entry is currently executing, it is aborted.
--------------------	---

restrictions	You must have ABORT_JOB access to the Entry to be deleted.
---------------------	--

related commands	• SET ENTRY
-------------------------	---

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the deletion of each Entry which matches your selection specification. <i>/FIRST</i> If specified, deletes only the first entry which matches the selection criteria. The entries which match the selection criteria are sorted by scheduled time, priority and submit time and the first one is selected for deletion. <i>/LOG</i> This qualifier can be used to specify that each deletion should be logged by displaying a message on SYS\$OUTPUT. <i>/SUBMITTED_BY=username</i> Selects only entries which were submitted by the specified username. <i>/SYSTEM_ID=system-id</i> Selects only entries which are in the specified <i>JAMS</i> System.
-------------------	--

DELETE JOB

Deletes one or more *JAMS* Job Definitions.

FORMAT	DELETE JOB <i>job-name</i>
---------------	-----------------------------------

PARAMETERS	<i>job-name</i> A valid <i>JAMS</i> Job name or wildcard specification.
-------------------	---

DESCRIPTION	The DELETE JOB command is used to delete one or more <i>JAMS</i> Job Definitions.
--------------------	---

restrictions	You must have DELETE access to the Job to be deleted.
---------------------	---

related commands	• DEFINE JOB • EXTRACT JOB
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the deletion of each Job which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each deletion should be logged by displaying a message on SYS\$OUTPUT.
-------------------	---

DELETE MENU

DELETE MENU

Deletes one or more *JAMS* Menu Definitions.

FORMAT	DELETE MENU <i>menu-name</i>
---------------	-------------------------------------

PARAMETERS	<i>menu-name</i> A valid <i>JAMS</i> Menu name or wildcard specification.
-------------------	---

DESCRIPTION	The DELETE MENU command is used to delete one or more <i>JAMS</i> Menu Definitions.
--------------------	---

restrictions	You must have DELETE access to the Menu to be deleted.
---------------------	--

related commands	• DEFINE MENU • EXTRACT MENU
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the deletion of each Menu which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each deletion should be logged by displaying a message on SYSS\$OUTPUT.
-------------------	---

DELETE METHOD

Deletes one or more *JAMS* Methods.

FORMAT	DELETE METHOD <i>method-name</i>
---------------	---

PARAMETERS	<i>method-name</i> A valid <i>JAMS</i> Method name or wildcard specification.
-------------------	---

DESCRIPTION	The DELETE METHOD command is used to delete one or more <i>JAMS</i> Method Definitions.
--------------------	---

restrictions	You must have EXECUTE access to <i>JAMS</i> Configuration.
---------------------	--

related commands	• CREATE METHOD • EXTRACT METHOD
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the deletion of each Method which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each deletion should be logged by displaying a message on SYS\$OUTPUT.
-------------------	--

DELETE RESOURCE

DELETE RESOURCE

Deletes one or more *JAMS* Resource Definitions.

FORMAT	DELETE RESOURCE <i>resource-name</i>
---------------	---

PARAMETERS	<i>resource-name</i> A valid <i>JAMS</i> Resource name or wildcard specification.
-------------------	---

DESCRIPTION	The DELETE RESOURCE command is used to delete one or more <i>JAMS</i> Resource Definitions.
--------------------	---

restrictions	You must have DELETE access to Resource Definitions.
---------------------	--

related commands	• CREATE RESOURCE • EXTRACT RESOURCE
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the deletion of each Resource which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each deletion should be logged by displaying a message on SYS\$OUTPUT.
-------------------	--

DELETE SETUP

Deletes one or more *JAMS* Setup Definitions.

FORMAT	DELETE SETUP <i>setup-name</i>
---------------	---------------------------------------

PARAMETERS	<i>setup-name</i> A valid <i>JAMS</i> Setup name or wildcard specification.
-------------------	---

DESCRIPTION	The DELETE SETUP command is used to delete one or more <i>JAMS</i> Setup Definitions.
--------------------	---

restrictions	You must have DELETE access to the Setup to be deleted.
---------------------	---

related commands	• DEFINE SETUP • EXTRACT SETUP
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the deletion of each Setup which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each deletion should be logged by displaying a message on SYS\$OUTPUT.
-------------------	---

DELETE SYSTEM

DELETE SYSTEM

Deletes one or more *JAMS* System Definitions.

FORMAT	DELETE SYSTEM <i>system-id</i>
---------------	---------------------------------------

PARAMETERS	<i>system-id</i> A valid <i>JAMS</i> System I.D. or wildcard specification.
-------------------	---

DESCRIPTION	The DELETE SYSTEM command is used to delete one or more <i>JAMS</i> System Definitions.
--------------------	---

restrictions	You must have DELETE access to the System to be deleted.
---------------------	--

related commands	• DEFINE SYSTEM • EXTRACT SYSTEM
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the deletion of each System which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each deletion should be logged by displaying a message on SYS\$OUTPUT.
-------------------	--

DELETE TIME

Deletes one or more *JAMS* Named Times.

FORMAT	DELETE TIME <i>time-name</i>
---------------	-------------------------------------

PARAMETERS	<i>time-name</i> A valid <i>JAMS</i> Time name or wildcard specification.
-------------------	---

DESCRIPTION	The DELETE TIME command is used to delete one or more <i>JAMS</i> Named Time Definitions.
--------------------	---

restrictions	You must have DELETE access to Named Time Definitions.
---------------------	--

related commands	• CREATE TIME • EXTRACT TIME
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the deletion of each Named Time which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each deletion should be logged by displaying a message on SYS\$OUTPUT.
-------------------	--

DELETE TRIGGER

DELETE TRIGGER

Deletes one or more *JAMS* Trigger Definitions.

FORMAT	DELETE TRIGGER <i>trigger-name</i>
---------------	---

PARAMETERS	<i>trigger-name</i> A valid <i>JAMS</i> Trigger name or wildcard specification.
-------------------	---

DESCRIPTION	The DELETE TRIGGER command is used to delete one or more <i>JAMS</i> Trigger Definitions.
--------------------	---

restrictions	You must have DELETE access to the Trigger Definitions.
---------------------	---

related commands	• DEFINE TRIGGER • EXTRACT TRIGGER
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the deletion of each Trigger which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each deletion should be logged by displaying a message on SYS\$OUTPUT.
-------------------	---

DELETE VARIABLE

Deletes one or more *JAMS* Variables.

FORMAT	DELETE VARIABLE <i>variable-name</i>
---------------	---

PARAMETERS	<i>variable-name</i> A valid <i>JAMS</i> Variable name or wildcard specification.
-------------------	---

DESCRIPTION	The DELETE VARIABLE command is used to delete one or more <i>JAMS</i> Variables.
--------------------	--

restrictions	You must have DELETE access to the Variable to be deleted.
---------------------	--

related commands	• CREATE VARIABLE • GET VARIABLE • SET VARIABLE • SHOW VARIABLE
-------------------------	---

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the deletion of each Variable which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each deletion should be logged by displaying a message on SYS\$OUTPUT.
-------------------	--

DISABLE TIME

DISABLE TIME

Disables a Named Time.

FORMAT	DISABLE TIME <i>time-name</i>
---------------	--------------------------------------

PARAMETERS	<i>time-name</i> A valid <i>JAMS</i> Named Time.
-------------------	--

DESCRIPTION	The DISABLE TIME command is used to disable a Time which is manually disabled. It can also be used to disable an automatically disabled time.
--------------------	--

restrictions	You must have MANAGE access to Time Definitions to use this command.
---------------------	---

related commands	• CREATE TIME • ENABLE TIME • EXTRACT TIME • SHOW TIME
-------------------------	--

DISABLE TRIGGER

Disables a Trigger.

FORMAT	DISABLE TRIGGER <i>trigger-name</i>
---------------	--

PARAMETERS	<i>trigger-name</i> A valid <i>JAMS</i> Trigger name.
-------------------	---

DESCRIPTION	The DISABLE TRIGGER command can be used to disable a Trigger which will prevent the Trigger from being fired until the Trigger is enabled with the ENABLE TRIGGER command.
--------------------	--

Disabling and enabling a Trigger will set the Trigger's last reset date to the current date and time.

restrictions	You must have MANAGE access to Trigger Definitions to use this command.
---------------------	---

related commands	• ENABLE TRIGGER • RESET TRIGGER • SHOW TRIGGER
-------------------------	---

ENABLE TIME

ENABLE TIME

Enables a Named Time.

FORMAT	ENABLE TIME <i>time-name</i>
---------------	-------------------------------------

PARAMETERS	<i>time-name</i> A valid <i>JAMS</i> Named Time.
-------------------	--

DESCRIPTION	The ENABLE TIME command is used to enable a Time which is manually enabled. It can also be used to enable an automatically enabled time.
--------------------	---

restrictions	You must have MANAGE access to Time Definitions to use this command.
---------------------	---

related commands	• CREATE TIME • DISABLE TIME • EXTRACT TIME • SHOW TIME
-------------------------	---

ENABLE TRIGGER

Enables a Trigger which has been disabled.

FORMAT	ENABLE TRIGGER <i>trigger-name</i>
---------------	---

PARAMETERS	<i>trigger-name</i> A valid <i>JAMS</i> Trigger name.
-------------------	---

DESCRIPTION	The ENABLE TRIGGER command is used to enable a Trigger which has been disabled with the DISABLE TRIGGER command. A Trigger will not be fired if it has been disabled.
--------------------	---

When you enable a Trigger, the Trigger's last reset date is set to the current date and time.

restrictions	You must have MANAGE access to Trigger Definitions to use this command.
---------------------	--

related commands	• DISABLE TRIGGER • RESET TRIGGER • SHOW TRIGGER
-------------------------	---

EVALUATE DATE

Evaluates an English language date specification.

FORMAT	EVALUATE DATE <i>date-specification</i>
---------------	--

PARAMETERS	<i>date-specification</i> A valid <i>JAMS</i> English language date specification. If the specification includes embedded spaces it must be enclosed in quotes.
-------------------	---

DESCRIPTION	The EVALUATE DATE command is used to parse an English language date specification and return the absolute date which the phrase represents. This command can be used to test your date specifications and from within your DCL routines to enhance your applications. There is also a callable version of this command.
--------------------	--

QUALIFIERS	<i>/GLOBAL</i> Using this qualifier causes the returned DCL symbol to be defined in the global symbol table. <i>/LOCAL</i> Using this qualifier causes the returned DCL symbol to be defined in the local symbol table. <i>/SYMBOL=symbol_name</i> If specified, the resulting date will be placed in the specified DCL symbol. If you do not use this qualifier, the resulting date is displayed on your terminal. <i>/TODAY_IS=date</i> The date specification will be interpreted as if today was the date specified with this qualifier. If you do not use this qualifier, the default is the current system date.
-------------------	---

EXIT

Exits from the *JAMS_MASTER* program.

FORMAT

EXIT

DESCRIPTION

The *EXIT* command exits the *JAMS_MASTER* program and returns control to the environment which invoked *JAMS_MASTER*.

EXTRACT DATE

Extract one or more Date Definitions.

FORMAT	EXTRACT DATE
--------	--------------

DESCRIPTION	The EXTRACT DATE command is used to extract one or more Date Definitions. This definition can be used to replicate the Dates on another VMSccluster.
--------------------	--

restrictions	You must have INQUIRE access to Date definitions.
---------------------	---

related commands	• CREATE DATE • CREATE DTYPE • EXTRACT DTYPE
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the extraction of each Date which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT. <i>/OUTPUT=file_name</i> Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT. <i>/SINCE=date</i> If specified, only dates on or after the specified date will be extracted. <i>/TYPE=date-type</i> If specified, only dates for the specified Date Type will be extracted.
-------------------	--

EXTRACT DTYPE

Extract one or more Date Type Definitions.

FORMAT **EXTRACT DTYPE** *date-type-name*

PARAMETERS ***date-type-name***
 The Name of the Date Type which you want to extract. You can use wildcards to extract more than one definition.

DESCRIPTION The EXTRACT DTYPE command is used to extract a complete Date Type Definition. This definition can be used to replicate the Date Type on another VMScluster.

restrictions You must have INQUIRE access to Date Type definitions.

related commands

- CREATE DATE
- CREATE DTYPE
- EXTRACT DATE

QUALIFIERS ***/CONFIRM***
 This qualifier is used to specify that you want to be prompted to confirm the extraction of each Date Type which matches your selection specification.

/LOG
 This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT.

/OUTPUT=file_name
 Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.

EXTRACT JOB

Extract one or more Job Definitions.

FORMAT	EXTRACT JOB <i>job-name</i>
---------------	------------------------------------

PARAMETERS	<i>job-name</i>
-------------------	------------------------

The Job Name of the Job which you want to extract. You can use wildcards to extract more than one definition.

DESCRIPTION	The EXTRACT JOB command is used to extract a complete Job Definition. This definition can be used to replicate the Job on another VMScluster.
--------------------	---

restrictions	You must have INQUIRE access to the Job whose definition you are trying to obtain.
---------------------	--

related commands	• DEFINE JOB
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i>
-------------------	------------------------

This qualifier is used to specify that you want to be prompted to confirm the extraction of each Job which matches your selection specification.

	<i>/LOG</i>
--	--------------------

This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT.

	<i>/OUTPUT=file_name</i>
--	---------------------------------

Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.

	<i>/SOURCE</i>
--	-----------------------

Specifies that the job's source code should be included in the definition. /SOURCE is the default. Specify /NOSOURCE if you do not want the source included in the definition.

	<i>/RECOVERY</i>
--	-------------------------

Specifies that the job's recovery instructions should be included in the definition. /RECOVERY is the default. Specify /NORECOVERY if you do not want the recovery instructions included in the definition.

	<i>/SYSTEM=system_id</i>
--	---------------------------------

Only jobs which match the Job Name specification and are in the JAMS System specified with this qualifier will be extracted.

EXTRACT MENU

Extract one or more Menu Definitions.

FORMAT **EXTRACT MENU** *menu-name*

PARAMETERS ***menu-name***
 The Name of the Menu which you want to extract. You can use wildcards to extract more than one definition.

DESCRIPTION The EXTRACT MENU command is used to extract a complete Menu Definition. This definition can be used to replicate the Menu on another VMScluster.

restrictions You must have INQUIRE access to Menu definitions.

related commands • DEFINE MENU

QUALIFIERS ***/CONFIRM***
 This qualifier is used to specify that you want to be prompted to confirm the extraction of each Menu which matches your selection specification.

/LOG
 This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT.

/OUTPUT=file_name
 Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.

EXTRACT METHOD

EXTRACT METHOD

Extract one or more Method Definitions.

FORMAT	EXTRACT METHOD <i>method-name</i>
---------------	--

PARAMETERS	<i>method-name</i> The Name of the Method which you want to extract. You can use wildcards to extract more than one definition.
-------------------	---

DESCRIPTION	The EXTRACT METHOD command is used to extract a complete Method Definition. This definition can be used to replicate the Method on another system.
--------------------	--

restrictions	You must have EXECUTE access to JAMS Configuration.
---------------------	---

related commands	• CREATE METHOD • DELETE METHOD
-------------------------	---

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the extraction of each Method which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT. <i>/OUTPUT=file_name</i> Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.
-------------------	---

EXTRACT RESOURCE

Extract one or more Resource Definitions.

FORMAT **EXTRACT RESOURCE** *resource-name*

PARAMETERS ***resource-name***
 The Name of the Resource which you want to extract. You can use wildcards to extract more than one definition.

DESCRIPTION The EXTRACT RESOURCE command is used to extract a complete Resource Definition. This definition can be used to replicate the Resource on another system.

restrictions You must have INQUIRE access to Resource Definitions.

related commands

- ACQUIRE RESOURCE
- CREATE RESOURCE
- DELETE RESOURCE
- RELEASE RESOURCE

QUALIFIERS ***/CONFIRM***
 This qualifier is used to specify that you want to be prompted to confirm the extraction of each Resource which matches your selection specification.

/LOG
 This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT.

/OUTPUT=file_name
 Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.

EXTRACT SECURITY

Extract one or more JAMS Security Access Control Lists.

FORMAT **EXTRACT SECURITY** *security-name*

PARAMETERS ***security-name***

The name of the JAMS Security ACL which is to be extracted. If you do not supply a Security Name, all of the ACL's are extracted. Valid names are:

CONFIGURATION	Access to JAMS configuration.
DATES	Access to Date definitions
DTYPES	Access to Date Type definitions.
HISTORY	Access to History Inquiry.
JOBS	Access to Job definitions.
MENUS	Access to Menu definitions
MONITOR	Access to Job Monitor.
NODES	Access to Node definitions.
RESOURCE	Access to Resource definitions.
SECURITY	Access to Security settings.
SETUPS	Access to Setup definitions.
SYSTEMS	Access to System definitions.
TIMES	Access to Named Time definitions.
TRIGGERS	Access to Trigger definitions
USERNAMES	Access to Username access control.
VARIABLES	Access to Variable definitions.

DESCRIPTION The EXTRACT SECURITY command is used to extract a complete Security Definition. This definition can be used to replicate the ACL on another VMScluster.

restrictions You must have INQUIRE access to Access Control.

related commands • SET SECURITY

QUALIFIERS***/CONFIRM***

This qualifier is used to specify that you want to be prompted to confirm the extraction of each ACL.

/LOG

This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT.

/OUTPUT=file_name

Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.

EXTRACT SETUP

Extract one or more Setup Definitions.

FORMAT	EXTRACT SETUP <i>setup-name</i>
---------------	--

PARAMETERS	<i>setup-name</i>
-------------------	--------------------------

The Setup Name of the Setup which you want to extract. You can use wildcards to extract more than one definition.

DESCRIPTION	
--------------------	--

The EXTRACT SETUP command is used to extract a complete Setup Definition. This definition can be used to replicate the Setup on another VMScluster.

restrictions	
---------------------	--

You must have INQUIRE access to the Setup whose definition you are trying to obtain.

related commands	
-------------------------	--

- DEFINE SETUP

QUALIFIERS	
-------------------	--

/CONFIRM

This qualifier is used to specify that you want to be prompted to confirm the extraction of each Setup which matches your selection specification.

/JOB=job_name

Only Setups which match the Setup Name specification and execute the JAMS Job specified with this qualifier will be extracted.

/LOG

This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT.

/OUTPUT=file_name

Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.

/RECOVERY

Specifies that the Setups recovery instructions should be included in the definition. /RECOVERY is the default. Specify /NORECOVERY if you do not want the recovery instructions included in the definition.

/SYSTEM=system_id

Only Setup which match the Setup Name specification and are in the JAMS System specified with this qualifier will be extracted.

EXTRACT SYSTEM

Extract one or more System Definitions.

FORMAT	EXTRACT SYSTEM <i>system-id</i>
---------------	--

PARAMETERS	<i>system-id</i> The System I.D. of the System which you want to extract. You can use wildcards to extract more than one definition.
-------------------	--

DESCRIPTION	The EXTRACT SYSTEM command is used to extract a complete System Definition. This definition can be used to replicate the System on another VMScluster.
--------------------	--

restrictions	You must have INQUIRE access to the System whose definition you are trying to obtain.
---------------------	---

related commands	• DEFINE SYSTEM
-------------------------	---

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the extraction of each System which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT. <i>/OUTPUT=file_name</i> Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT. <i>/RECOVERY</i> Specifies that the Systems recovery instructions should be included in the definition. /RECOVERY is the default. Specify /NORECOVERY if you do not want the recovery instructions included in the definition.
-------------------	---

EXTRACT TIME

Extract one or more Named Time Definitions.

FORMAT	EXTRACT TIME <i>time-name</i>
---------------	--------------------------------------

PARAMETERS	<i>time-name</i>
-------------------	-------------------------

The Name of the Time which you want to extract. You can use wildcards to extract more than one definition.

DESCRIPTION	The EXTRACT TIME command is used to extract a complete Named Time Definition. This definition can be used to replicate the Time on another VMScluster.
--------------------	--

restrictions	You must have INQUIRE access to Named Times.
---------------------	--

related commands	
-------------------------	--

- CREATE TIME
- DISABLE TIME
- ENABLE TIME
- SHOW TIME

QUALIFIERS	<i>/CONFIRM</i>
-------------------	------------------------

This qualifier is used to specify that you want to be prompted to confirm the extraction of each Time which matches your selection specification.

/LOG

This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT.

/OUTPUT=file_name

Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.

EXTRACT TRIGGER

Extract one or more Trigger Definitions.

FORMAT **EXTRACT TRIGGER** *trigger-name*

PARAMETERS *trigger-name*
 The Name of the Trigger which you want to extract. You can use wildcards to extract more than one definition.

DESCRIPTION The EXTRACT TRIGGER command is used to extract a complete Trigger Definition. This definition can be used to replicate the Trigger on another VMScluster.

restrictions You must have INQUIRE access to Trigger Definitions.

related commands

- DEFINE TRIGGER
- SHOW TRIGGER

QUALIFIERS ***/CONFIRM***
 This qualifier is used to specify that you want to be prompted to confirm the extraction of each Trigger which matches your selection specification.

/LOG
 This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT.

/OUTPUT=file_name
 Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.

EXTRACT USERNAME

EXTRACT USERNAME

Extract one or more Username Access Control Definitions.

FORMAT	EXTRACT USERNAME <i>username</i>
---------------	---

PARAMETERS	<i>username</i> The OpenVMS Username which you want to extract. You can use wildcards to extract more than one definition.
-------------------	--

DESCRIPTION	The EXTRACT USERNAME command is used to extract a complete Username Access Control Definition. This definition can be used to replicate the Username Access Definition on another VMSccluster.
--------------------	--

restrictions	You must have INQUIRE access to Username Security.
---------------------	--

related commands	• SET USERNAME
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the extraction of each Username which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT. <i>/OUTPUT=file_name</i> Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.
-------------------	---

EXTRACT VARIABLE

Extract one or more Variable Definitions.

FORMAT **EXTRACT VARIABLE** *variable-name*

PARAMETERS ***variable-name***
 The Name of the Variable which you want to extract. You can use wildcards to extract more than one definition.

DESCRIPTION The EXTRACT VARIABLE command is used to extract a complete Variable Definition. This definition can be used to replicate the Variable on another VMScluster.

restrictions You must have INQUIRE access to the Variable whose definition you are trying to obtain.

related commands • CREATE VARIABLE

QUALIFIERS ***/CONFIRM***
 This qualifier is used to specify that you want to be prompted to confirm the extraction of each Variable which matches your selection specification.

/LOG
 This qualifier can be used to specify that each extraction should be logged by displaying a message on SYS\$OUTPUT.

/OUTPUT=file_name
 Specifies the name of a file where the definitions will be written. If you do not specify the /OUTPUT qualifier, the definitions are written to SYS\$OUTPUT.

FAKE COMPLETION—Fake a Job Completion

Generates a fake Job Completion. A History record is written, dependencies are satisfied and Trigger Events are updated.

FORMAT	FAKE COMPLETION <i>job-name</i>
---------------	--

PARAMETERS	<i>job-name</i> A valid JAMS Job name.
-------------------	--

DESCRIPTION	This command is used to fake a Job Completion. It is useful when a Job has failed but does not need to be re-run, you can fake a successful completion to release dependent Jobs.
--------------------	---

restrictions	You must have SUBMIT access to the specified Job.
---------------------	---

QUALIFIERS	<i>/SEVERITY=severity</i> You can use the /SEVERITY qualifier to specify the severity of the final status of the faked completion. Valid values for severity are: SUCCESS INFORMATIONAL WARNING ERROR FATAL
-------------------	--

GET STATUS

Gets a Jobs status text.

FORMAT	GET STATUS
--------	------------

DESCRIPTION	The <i>GET STATUS</i> command can be used when you want to retrieve a Job's status text. The Job whose status is retrieved is the one which executes the command.
-------------	---

related commands

- SET STATUS
-

QUALIFIERS

/GLOBAL

Using this qualifier causes the returned DCL symbol to be defined in the global symbol table.

/LOCAL

Using this qualifier causes the returned DCL symbol to be defined in the local symbol table.

/SYMBOL=DCL-symbol-name

This qualifier can be used to specify the name of a DCL symbol which will hold the retrieved status. If this qualifier is not used, the status is returned in a symbol named *JAMS_JOB_STATUS*.

GET VARIABLE

Put the current value of a Variable into a DCL Symbol.

FORMAT	GET VARIABLE <i>variable-name</i>
---------------	--

PARAMETERS	<i>variable-name</i> A valid <i>JAMS</i> Variable name. You can access a Variable on a remote node by specifying the remote node name and access control information in the standard DECnet format.
-------------------	---

DESCRIPTION	The GET VARIABLE command is used to get the current value of a <i>JAMS</i> Variable and place the value into a DCL symbol.
--------------------	--

restrictions	You must have INQUIRE access to the Variable whose value you are trying to obtain.
---------------------	--

related commands	• CREATE VARIABLE • DELETE VARIABLE • SET VARIABLE • SHOW VARIABLE
-------------------------	--

QUALIFIERS	<i>/GLOBAL</i> Using this qualifier causes the returned DCL symbol to be defined in the global symbol table. <i>/LOCAL</i> Using this qualifier causes the returned DCL symbol to be defined in the local symbol table. <i>/SYMBOL=symbol_name</i> Specifies the name of the DCL symbol which will receive the value of the <i>JAMS</i> Variable. If you do not specify this qualifier, the name of the Variable will be used as the name of the DCL symbol.
-------------------	--

HELP

Provides help for the *JAMS_MASTER* command line environment.

FORMAT **HELP** *help-item*

DESCRIPTION The *HELP* command provides OpenVMS style help for the *JAMS_MASTER* command line environment.

PARAMETERS *help-item*
A partial or full string used to specify the help topic.

MENU

Invoke the screen based environment.

FORMAT

MENU

DESCRIPTION

The *MENU* command is used to enter the full-screen, menu based mode. Refer to the *JAMS User Guide* for additional information on using the screen based environment.

restrictions

The screen based environment requires an SMG compatible terminal.

QUALIFIERS

/CONNECTIONS (default)

/NOCONNECTIONS

The */CONNECTIONS* qualifier enables the Gold/C function key to allow a user to establish connections to remote nodes. The function key is available from History Inquiry, Job Submission and Monitor Current Jobs.

/DEBUG

/NODEBUG (default)

The */DEBUG* qualifier specifies that Jobs should be submitted in debug mode. This mode is used by programmers to test and debug Jobs. When a Job is submitted in debug mode, it is always submitted under the OpenVMS username of the person making the request. If the System definition specifies a OpenVMS username, it is ignored.

When submitting a Job in debug mode, the logical names used during Job parsing do not need to be defined at the executive level. This enables a programmer to define logical names in their process specific logical name tables which will direct the JAMS parsing routines to a test version of the Job source files and/or the Systems template library.

/MENU=menu_name

The */MENU* qualifier is used to specify a valid Menu Name which will be used to build the initial menu when the Job Submission option is selected.

The default is to search for the logical name JAMS_DEFAULT_MENU which specifies the top level Menu Name. If this logical is not defined, or defined incorrectly, System based menus will be used when submitting a job.

/OPTION=menu_option

Bypasses the *JAMS* Main Menu and invokes the specified menu option. The valid values for the menu option are:

- ACCESS - Security Maintenance
- CONFIGURATION - Configuration Maintenance
- DATES - Date Definitions
- DTYPES - Date Type Definitions

HISTORY - History Inquiry
 JOBS - Job Definitions
 MENUS - Menu Definitions
 MONITOR - Monitor Current Jobs
 SETUPS - Setup Definitions
 SHOW_JOBS - Show Current Jobs
 SUBMIT - Job Submission Menu
 SYSTEMS - System Definitions
 TRIGGER - Trigger Definitions
 VARIABLE - Variable Definitions

/PROMPT (default)

/NOPROMPT

If you specify /NOPROMPT, you will not be prompted for the values for a Job's parameters or report overrides. Default values are taken instead.

/SET_SYMBOLS

/NOSET_SYMBOLS (default)

If you specify /SET_SYMBOLS, a DCL symbol will be defined, or updated, for each of the Job's parameters. The symbol name will be the same as the parameter name, and the value will be the value used when submitting the Job.

If a symbol is being created, it is created in the local symbol table. If it is being updated, it is updated in whichever table it is currently defined.

/SUBMIT (default)

/NOSUBMIT

The /NOSUBMIT qualifier is useful when you are testing a Job definition or a template library. If you specify /NOSUBMIT, the job is not submitted to a batch queue. Instead, a file is created in the current default directory with the name of the Job and an extension of .JAMS. This file contains the parsed Job Source file exactly as it would be submitted to a batch queue.

/SYSTEM[=system_id]

The /SYSTEM qualifier is used to specify that System based menus should be used when the Job Submission option is selected. A single level menu will be built which contains all of the Job and Setup definitions for the specified System.

If you do not specify a System I.D. all of the available Systems are presented as the top level menu.

/USE_SYMBOLS (default)

/NOUSE_SYMBOLS

If the /USE_SYMBOLS qualifier is specified, either explicitly or by default, when JAMS is building a parameter screen, it will use the value of DCL symbols whose names match a parameter's name for the parameter's default value.

When JAMS determines the default value for a parameter, it takes the following steps:

- 1 Use the default value entered in the parameter definition.

MENU

- 2 If a Setup is being submitted, use the default value from the Setup's parameter screen.
- 3 If /USE_SYMBOLS is specified, see if there is a DCL symbol with the same name as the parameter and use the symbol's value, if defined.

EXAMPLES

JAMS> MENU/MENU=TOP_PAYROLL

In this example, the user will be placed at the main menu of the screen based environment. If they select the *Job Submission* menu option, the menu displayed will be the menu named TOP_PAYROLL.

JAMS> MENU/MENU=TOP_PAYROLL/OPTION=SUBMIT

In this example, the user will be placed in the Job Submission menu option of the screen based environment. The menu displayed will be the menu named TOP_PAYROLL.

JAMS> MENU/DEBUG

In this example, the user will be placed at the main menu of the screen based environment. Any jobs submitted with the *Job Submission* menu option will be submitted in debug mode.

RECORD

Record an interactive session for playback in batch mode.

FORMAT	RECORD <i>session-filespec</i>
---------------	---------------------------------------

PARAMETERS	<i>session-filespec</i> A file specification where the recorded session will be written.
-------------------	--

DESCRIPTION	The RECORD command is used to record an interactive session. The recorded session can then be used as the source for a job that uses the PLAY execution method. A recorded session file is an ASCII text file that you can edit with a text editor. After recording a session, you may want to edit the session file to replace entered values with <<param-name>> parameter references.
--------------------	--

RELEASE RESOURCE

Release a quantity of a *JAMS* Resource.

FORMAT	RELEASE RESOURCE <i>resource-name [quantity]</i>
---------------	---

PARAMETERS	<i>resource-name</i> A valid <i>JAMS</i> Resource name.
	<i>quantity</i> The amount of the resource to be released. The default to to release all that has been acquired.

DESCRIPTION	The RELEASE RESOURCE command is used to release a quantity of a specific <i>JAMS</i> Resource.
--------------------	--

related commands

- ACQUIRE RESOURCE
- CREATE RESOURCE
- DELETE RESOURCE
- SET RESOURCE
- SHOW RESOURCE

RENEW LOG_FILE

Creates a new JAMS.LOG log file.

FORMAT

RENEW LOG_FILE

DESCRIPTION

The *JAMS* Monitor, Schedule and Network processes write status messages to a log file. This file is named JAMS.LOG and is located in the JAMS_DATA: directory. You can use the *RENEW LOG_FILE* command to tell all of the *JAMS* processes, cluster-wide, to close the current log file and open a new one.

At any time, you can check the contents of the *JAMS* log file with the OpenVMS *TYPE* command.

Normally you do not need to issue this command because the *JAMS_PURGE* job, which runs once a week, renews the log and purges the file back to three versions.

RESET JOB STATISTICS

RESET JOB STATISTICS

Resets a Job's execution statistics.

FORMAT	RESET JOB STATISTICS <i>job-name</i>
---------------	---

PARAMETERS	<i>job-name</i> A valid <i>JAMS</i> Job name, wildcards are allowed.
-------------------	--

DESCRIPTION	The RESET JOB STATISTICS command is used to reset the execution statistics of one or more Jobs. All statistics are set to zero.
--------------------	---

restrictions	You must have CHANGE access to Job Definitions and to the specific Jobs which you are resetting.
---------------------	--

related commands	• RESET SETUP STATISTICS
-------------------------	--

QUALIFIERS	<i>/CONFIRM</i> This qualifier is used to specify that you want to be prompted to confirm the resetting of each Job which matches your selection specification. <i>/LOG</i> This qualifier can be used to specify that each reset should be logged by displaying a message on SYS\$OUTPUT. <i>/SYSTEM=system_id</i> This qualifier can be used to specify a <i>JAMS</i> System I.D. which is used to select Jobs which will be reset. Wildcards are allowed.
-------------------	--

RESET SETUP STATISTICS

Resets a Setup's execution statistics.

FORMAT **RESET SETUP STATISTICS** *setup-name*

PARAMETERS ***setup-name***
A valid *JAMS* Setup name, wildcards are allowed.

DESCRIPTION The RESET SETUP STATISTICS command is used to reset the execution statistics of one or more Setups. All statistics are set to zero.

restrictions You must have CHANGE access to Setup Definitions and to the specific Setups which you are resetting.

related commands • RESET JOB STATISTICS

QUALIFIERS ***/CONFIRM***
This qualifier is used to specify that you want to be prompted to confirm the resetting of each Setup which matches your selection specification.

/JOB_NAME=job_name
This qualifier can be used to specify a *JAMS* Job Name which is used to select Setups which will be reset. Only Setups which match the Setup specified and run a Job which matches the Job specified will be reset. Wildcards are allowed.

/LOG
This qualifier can be used to specify that each reset should be logged by displaying a message on SYS\$OUTPUT.

/SYSTEM=system_id
This qualifier can be used to specify a *JAMS* System I.D. which is used to select Setups which will be reset. Wildcards are allowed.

RESET TRIGGER

RESET TRIGGER

Resets a Trigger after it has been fired.

FORMAT	RESET TRIGGER <i>trigger-name</i>
---------------	--

PARAMETERS	<i>trigger-name</i> A valid <i>JAMS</i> Trigger name.
-------------------	---

DESCRIPTION	The RESET TRIGGER command is used to reset a Trigger after it has been fired. When all of a Trigger's events are true, the Trigger will fire and all of the Trigger's actions will be performed. If the Trigger is set to be reset when it is fired, the Trigger is automatically reset. Otherwise, the Trigger is marked as fired and it will not be fired again until it has been reset with this command.
--------------------	---

When a Trigger has a Job Completion event, the specified Job must complete since the Trigger's Last Reset date and time. This date and time is set when the Trigger is reset and when a disabled Trigger is enabled.

restrictions	You must have RESET access to Trigger Definitions to use this command.
---------------------	--

related commands	• DISABLE TRIGGER • ENABLE TRIGGER
-------------------------	--

SET AGENT

Set attributes for communicating with *JAMS* Agents.

FORMAT

SET AGENT

DESCRIPTION

This command is used to set attributes used when communicating with a remote *JAMS* Agent.

QUALIFIERS

/PORT=tcpip-port

Specifies the TCP/IP port used to communicate with the *JAMS* Agent. The default is port 77 which is a well known port assigned by the Internet Assigned Numbers Authority (IANA) to “any private RJE server”.

/SECRET=secret

/NOSECRET

Specifies the secret used to authenticate a remote *JAMS* Agent. The default is no secret which means that authentication isn’t enabled. Note that this is the *JAMS* Requester to *JAMS* Agent authentication. Even if this authentication is disabled, the Requester must still supply a valid username/password combination when making a request.

SET ENTRY

Changes the current status or attributes of one or more Jobs in the *JAMS* schedule.

FORMAT	SET ENTRY <i>[job-name entry-number]</i>
---------------	---

PARAMETERS	<i>job-name entry-number</i>
-------------------	---------------------------------------

A Job Name or *JAMS* Entry number. Wildcards are allowed when specifying a Job Name.

Note that if an entry number is specified, it must be the *JAMS* Entry number, not an OpenVMS queue entry number.

DESCRIPTION	This command is used to change the current status or attributes of one or more Jobs in the <i>JAMS</i> schedule.
--------------------	--

QUALIFIERS	<i>/AFTER=time</i>
-------------------	---------------------------

/NOAFTER

Modifies the Jobs scheduled time. Specify */NOAFTER* to remove the schedule time.

/CONFIRM

/NOCONFIRM

Controls whether a request is made to confirm the change to each job which matches the selection criteria.

/FIRST

If specified, only the first job which matches the selection criteria is modified.

/FORCE

The */FORCE* qualifier is equivalent to */NOAFTER /NOHOLD /NODEPENDENCIES /NOPRECHECK /NOWINDOW /NOREQUIREMENT /NOSTEP*. In addition, if future versions of *JAMS* add features which could prevent a job from running, they will be removed by the */FORCE* qualifier.

/HOLD

/NOHOLD

Controls the hold status of the Job.

/LOG

/NOLOG

Controls whether the *SET ENTRY* command displays a message logging each job modified.

/NODEPENDENCIES

Releases the job from all unsatisfied dependencies.

/NOPRECHECK

Releases the Job from it's Precheck Job requirement.

/NOREQUIREMENTS

Releases the Job from it's Resource requirements.

/NOSTEP

Releases the Job from it's Setup Step requirement.

/NOWINDOW

Releases the Job from it's Time window requirement.

/PRIORITY=priority

Modifies the scheduling priority of the Job.

/QUEUE=new-queue

Moves the Job from its current queue to the specified queue.

/RELEASE

The /RELEASE qualifier is equivalent to /NOAFTER/NOHOLD.

/SUBMITTED_BY[=username]

Selects only those jobs which were submitted by the specified user. If you use this qualifier without specifying a username, the username defaults to the current username.

/SYSTEM=system-id

Selects only those jobs which are in the specified JAMS System.

SET LICENSE

SET LICENSE

Registers a usage based license.

FORMAT

SET LICENSE

DESCRIPTION

The SET LICENSE command registers the job limit portion of a usage based license. You must also register and load a usage based license PAK using the OpenVMS License Management Facility.

restrictions

You must have the SYSPRV privilege to use this command.

QUALIFIERS

/LIMIT=limit

Specifies the daily job limit for this license.

/TOKEN=token

Specifies the license token that this job limit is assigned to. This token *must* match the product token field of the JAMS License PAK.

/CHECKSUM=checksum

Specifies the license checksum.

SET RESOURCE

Updates the quantity available of a *JAMS* Resource.

FORMAT **SET RESOURCE** *resource-name*

PARAMETERS ***resource-name***
A valid *JAMS* Resource name.

DESCRIPTION The SET RESOURCE command is used to update the quantity available of a *JAMS* Resource. The value can be replaced or updated, depending upon what qualifiers are used.

restrictions You must have CHANGE access to the Resource whose value you are trying to change.

related commands

- ACQUIRE RESOURCE
- CREATE RESOURCE
- DELETE RESOURCE
- RELEASE RESOURCE
- SHOW RESOURCE

QUALIFIERS ***/AVAILABLE=qty-available***
Sets the quantity available to the specified value.

/DECREMENT
Decrements the quantity available.

/INCREMENT
Increments the quantity available.

/NODE=nodename
Used to specify different available quantities for individual nodes. The /NODE qualifier may be used only on resources created with the /BY_NODE qualifier.

SET SECURITY

Sets the Access Control List for a *JAMS* function.

FORMAT **SET SECURITY** *security-name*

PARAMETERS ***security-name***

The name of the *JAMS* Security ACL which is to be replaced. Valid names are:

CONFIGURATION	Access to <i>JAMS</i> configuration.
DATES	Access to Date definitions
DTYPES	Access to Date Type definitions.
HISTORY	Access to History Inquiry.
JOBS	Access to Job definitions.
MENUS	Access to Menu definitions
MONITOR	Access to Job Monitor.
NODES	Access to Node definitions.
RESOURCE	Access to Resource definitions.
SECURITY	Access to Security settings.
SETUPS	Access to Setup definitions.
SYSTEMS	Access to System definitions.
TIMES	Access to Named Time definitions.
TRIGGERS	Access to Trigger definitions
USERNAMES	Access to Username access control.
VARIABLES	Access to Variable definitions.

DESCRIPTION The *SET SECURITY* command is used to manage the Access Control Lists for the various *JAMS* functions.

related commands

- **EXTRACT SECURITY**

QUALIFIERS ***/ACL=acl-spec***

The */ACL* qualifier is used to specify the Access Control List for the specified function.

SET STATUS

Sets a job's status text.

FORMAT	SET STATUS <i>status-value</i>
---------------	---------------------------------------

PARAMETERS	<i>status-value</i> The value for the job's status text. Enclose this in quotes to preserve case and embedded spaces.
-------------------	---

DESCRIPTION	The <i>SET STATUS</i> command can be used when you want to set a job's status text. The job whose status is changed is the one which executes the command.
--------------------	--

related commands	• GET STATUS
-------------------------	--

QUALIFIERS	<i>/MONITOR</i> This qualifier can be used to specify that the status text should be displayed in the <i>JAMS</i> Job Monitor.
-------------------	--

SET USERNAME

SET USERNAME

Sets the Access Control List and password for a username.

FORMAT	SET USERNAME <i>username</i>
---------------	-------------------------------------

PARAMETERS	<i>Username</i>
-------------------	------------------------

The Username being updated. If you specify the /DEFAULT qualifier, do not specify a username.

DESCRIPTION	
--------------------	--

The *SET USERNAME* command is used to define Access Control Lists and passwords for Usernames. The ACLs are used to control who may specify the username as the username under which jobs should execute.

A user with the CMKRNL or BYPASS privilege can specify any valid username.

If specified, the password is used when running a job on a foreign system (such as Windows NT).

Note: When running jobs on Windows NT, the Windows NT accounts must have the “Log on as a batch job” user right. You can grant this right to individual accounts or to a Group which is then assigned to NT accounts.

QUALIFIERS	
-------------------	--

/ACL=acl-spec

The /ACL qualifier is used to specify the Access Control List for the specified username. This does *not* grant the specified username any rights, it controls who may specify this username in a System, Job or Setup definition as the username which the Job should run under.

/DEFAULT

The /DEFAULT qualifier is used to define an ACL which is applied for any username which is not specifically listed.

/PASSWORD=password

Specifies a password which is used to execute jobs on a foreign system. This password is not used on the local OpenVMS system.

Passwords are encrypted when they are stored and when they are sent to a JAMS Agent. The rexec protocol requires unencrypted passwords.

EXAMPLES

```
JAMS> SET USERNAME PAYBATCH/ACL=(IDENTIFIER=PAY_MGR,ACCESS=SUBMIT)
```

In this example, an ACL is defined for the username PAYBATCH. Any OpenVMS user who has the PAY_MGR rights identifier is allowed to specify the username PAYBATCH as the username under which a job should run.

```
JAMS> SET USERNAME NTBACKUP/ACL=(IDENTIFIER=BACKUP,ACCESS=SUBMIT) -  
_JAMS>/PASSWORD=CLEARTEXTPASSWORD
```

In this example, an ACL is defined for the username NTBACKUP. Any OpenVMS user who has the BACKUP rights identifier is allowed to specify the username NTBACKUP as the username under which a job should run. If this username is used on a job which runs on a foreign system, the password CLEARTEXTPASSWORD is sent to the foreign system.

```
JAMS> SET USERNAME/DEFAULT/ACL=(IDENTIFIER=[1,*],ACCESS=SUBMIT)
```

In this example, the default username ACL is defined. When *JAMS* checks to see if a user is allowed to use a given username, if the given username is not specifically listed, the default username ACL is used. In this example, only people in the [1,*] UIC group are allowed to specify usernames which do not have a specific ACL.

SET VARIABLE

SET VARIABLE

Updates the current value of a *JAMS* Variable.

FORMAT	SET VARIABLE <i>variable-name</i> [<i>new-value</i>]
---------------	---

PARAMETERS	<i>variable-name</i>
-------------------	-----------------------------

A valid *JAMS* Variable name. You can access a Variable on a remote node by specifying the remote node name and access control information in the standard DECnet format.

new-value

The new value for the specified Variable. Enclose this in quotes to preserve case and embedded spaces. This value is not allowed if you specify one of the valid qualifiers.

If the Variable has a datatype of DATE, the new value must be specified in DD-MMM-YYYY format. If the Variable has a datatype of TIME, the new value must be specified in HH:MM:SS, 24 hour format.

DESCRIPTION	The SET VARIABLE command is used to update the current value of a <i>JAMS</i> Variable. The value can be replaced or updated, depending upon what qualifiers are used. The specified data and qualifiers must be compatible with the Variable's data type.
--------------------	--

restrictions	You must have CHANGE access to the Variable whose value you are trying to change.
---------------------	---

related commands	• CREATE VARIABLE • DELETE VARIABLE • GET VARIABLE • SHOW VARIABLE
-------------------------	--

QUALIFIERS	<i>/CURRENT</i>
-------------------	------------------------

This qualifier may be used with Variables which have a data type of DATE, TIME or DATETIME. If this qualifier is used, the Variable is set to the current date and/or time.

/DECREMENT

This qualifier can be used with Variables which have a data type of INTEGER, FLOAT, DATE or DATETIME. For data types of INTEGER and FLOAT, the current value is decremented by 1. For data types of DATE and DATETIME, the value is decremented by 1 day.

//INCREMENT

This qualifier can be used with Variables which have a data type of INTEGER, FLOAT, DATE or DATETIME. For data types of INTEGER and FLOAT, the current value is incremented by 1. For data types of DATE and DATETIME, the value is incremented by 1 day.

SHOW ENTRY

Displays a list of current Jobs known to the JAMS Scheduling sub-system and any dependencies which a Job is waiting on.

FORMAT **SHOW ENTRY** *[job-name | entry-number]*

PARAMETERS ***job-name | entry-number***
A Job Name or JAMS entry number. Wildcards are allowed when specifying a Job Name.

DESCRIPTION This command is used to display a list of the Jobs which are currently in the JAMS schedule. A job is in the JAMS schedule if it was submitted by JAMS or is currently executing. Jobs which are submitted by some means other than JAMS will not be seen until they start executing.

QUALIFIERS ***/BY_JOB_STATUS=(job-status,...)***
Selects for display only those jobs with the specified status. Specify the status with one or more of the following keywords:

COMPLETE	Jobs which have completed.
DEPENDENCY	Jobs which are waiting for dependencies,
EXECUTING	Jobs which are executing.
HOLDING	Jobs which are held until released.
PENDING	Jobs which are pending.
PRECHECK	Jobs which are waiting for their Precheck Jobs to complete.
RETAINED	Jobs which have completed and are retained in the queue.
RUNAWAY	Jobs which have been declared runaway.
STALLED	Jobs which are stalled.
TIMED	Jobs which are waiting for their scheduled time.
TIME_SLOT	Jobs which are waiting for their Time Window to open.
WAITING	The same as specifying (DEPENDENCY, PRECHECK, TIME_SLOT)

/FULL
Displays all information about the Job.

/DEPENDENCIES
Show each Job's dependencies.

/OUTPUT=output-filespec

Controls where the output of the SHOW ENTRY command is sent. The default is SYS\$OUTPUT.

/SUBMITTED_BY[=username]

Selects for display only those jobs which were submitted by the specified user. If you use this qualifier without specifying a username, the username default to the current username.

/SYSTEM=system-id

Selects for display only those jobs which are in the specified JAMS System.

SHOW LICENSE

SHOW LICENSE

Displays information about the license key used to activate JAMS.

FORMAT

SHOW LICENSE

DESCRIPTION

The SHOW LICENSE command displays information about the active *JAMS* license.

QUALIFIERS

/COUNTS

Displays the number of jobs run by *JAMS*.

/FULL

When displaying counts, displays the job count for each day.

/OUTPUT=output-filespec

Controls where the output of the SHOW LICENSE command is sent. The default is SYS\$OUTPUT.

SHOW RESOURCE

Shows one or more Resources.

FORMAT **SHOW RESOURCE** *resource-name*

PARAMETERS ***resource-name***
The name of a valid *JAMS* Resource. Wildcards are allowed.

DESCRIPTION The SHOW RESOURCE command is used to show the current state of one or more *JAMS* Resources.

related commands

- ACQUIRE RESOURCE
- EXTRACT RESOURCE
- RELEASE RESOURCE

QUALIFIERS ***/FULL***
This qualifier specifies that all information on the Resource should be displayed.

SHOW STATUS

SHOW STATUS

Displays the status of the Monitor and Schedule processes.

FORMAT

SHOW STATUS

DESCRIPTION

This command is used to display the current status of the *JAMS* Monitor and Schedule processes in the VAXcluster. There is a one line status for every node in the cluster plus one line to display the status of the Schedule process.

restrictions

You must have the OPER privilege to use this command.

EXAMPLES

The following is an example of the output produced by the SHOW STATUS command.

```
JAMS Monitor is running on GRAPE, process ID is 2E600096
JAMS Monitor is running on APPLE, process ID is 2E800097
JAMS Monitor is running on STRAW, process ID is 2FA00055
JAMS Monitor is running on BERRY, process ID is 2FC00055
JAMS Monitor is not running on TRAFIC
JAMS Monitor is not running on TIGHT
JAMS Schedule is running on GRAPE, process ID is 2E600097
JAMS Network is running on GRAPE, process ID is 2E600098
```

In this example, the *JAMS* Monitor process is running on four out of six nodes in the VAXcluster. The Schedule and Network processes are also running.

SHOW TIME

Shows one or more Named Times.

FORMAT **SHOW TIME** *time-name*

PARAMETERS ***time-name***
A valid *JAMS* Named Time. Wildcards are allowed.

DESCRIPTION The SHOW TIME command is used to show Names Times.

related commands

- CREATE TIME
 - ENABLE TIME
 - EXTRACT TIME
 - DISABLE TIME
-

QUALIFIERS ***/FULL***
This qualifier specifies that all information on the Named Times should be displayed.

SHOW TRIGGER

SHOW TRIGGER

Shows one or more Triggers.

FORMAT	SHOW TRIGGER <i>trigger-name</i>
---------------	---

PARAMETERS	<i>trigger-name</i> The name of a valid <i>JAMS</i> Trigger. Wildcards are allowed.
-------------------	---

DESCRIPTION	The SHOW TRIGGER command is used to show the current state of <i>JAMS</i> Triggers.
--------------------	---

related commands

- RESET TRIGGER
- DISABLE TRIGGER
- ENABLE TRIGGER

QUALIFIERS	<i>/ACTIONS</i> <i>/NOACTIONS (default)</i> The <i>/ACTIONS</i> qualifier specifies that the Triggers actions should also be shown. <i>/DISABLED</i> The <i>/DISABLED</i> qualifier specifies that only Triggers which are disabled should be shown. The default to to show Triggers regardless of their current state. <i>/ENABLED</i> The <i>/ENABLED</i> qualifier specifies that only Triggers which are enabled should be shown. The default to to show Triggers regardless of their current state. <i>/EVENTS</i> <i>/NOEVENTS (default)</i> The <i>/EVENTS</i> qualifier specifies that the Triggers events should also be shown. <i>/FIRED</i> The <i>/FIRED</i> qualifier specifies that only Triggers which are fired should be shown. The default to to show Triggers regardless of their current state. <i>/FULL</i> This qualifier specifies that all information on the Trigger should be displayed.
-------------------	---

SHOW VARIABLE

Displays information about a JAMS Variable.

FORMAT **SHOW VARIABLE** *variable-name*

PARAMETERS ***variable-name***
A valid JAMS Variable name.

DESCRIPTION The SHOW VARIABLE command can be used to show a Variable's current value and attributes.

restrictions You must have INQUIRE access to the Variable whose value you are trying to see.

related commands

- CREATE VARIABLE
- DELETE VARIABLE
- GET VARIABLE
- SET VARIABLE

QUALIFIERS ***/FULL***
Displays all information about the Variable.

START MONITOR

Starts the JAMS Monitor process on the local node.

FORMAT

START MONITOR

DESCRIPTION

This command starts a *JAMS* Monitor process on the local node. If a Monitor process is already running, an error message is displayed and no action is taken.

Starting a Monitor process may also result in a *JAMS_SCHEDULE* process being started. If the *JAMS_SCHEDULE* process is not running on any of the nodes in the VAXcluster, and this node is allowed to run the Schedule process, the newly started Monitor process will create a Schedule process.

A node is allowed to run the Schedule process if the value of the logical name *JAMS_SCHED_WEIGHT* is either undefined, or defined to be a value other than 0 (zero).

This command is issued by the *JAMS* start-up procedure which should be executed during system start-up. Since the *JAMS_MONITOR* process is very stable, you probably will never need to issue this command.

restrictions

You must have the OPER privilege to start a JAMS Monitor process.

START NETWORK

Restarts the JAMS Network process after it has been stopped with the STOP NETWORK command.

FORMAT	START NETWORK
---------------	----------------------

DESCRIPTION	This command is used to restart a <i>JAMS</i> Network process after the process has been stopped with the STOP NETWORK command. When you issue this command, a Network process will be started on the node with the highest, non-zero value for the JAMS_NETWORK_WEIGHT logical name.
--------------------	---

Normally you will never need to use this command because the *JAMS* Network process is started by default.

restrictions	You must have the OPER privilege to start a JAMS Network process.
---------------------	---

START SCHEDULE

START SCHEDULE

Restarts the JAMS Schedule process after it has been stopped with the STOP SCHEDULE command.

FORMAT

START SCHEDULE

DESCRIPTION

This command is used to restart a *JAMS* Schedule process after the process has been stopped with the STOP SCHEDULE command. When you issue this command, a Schedule process will be started on the node with the highest, non-zero value for the JAMS_SCHED_WEIGHT logical name.

restrictions

You must have the OPER privilege to start a JAMS Schedule process.

STOP MONITOR

Stops the JAMS Monitor and Schedule processes on one or all nodes.

FORMAT	STOP MONITOR
--------	--------------

DESCRIPTION	This command stops the <i>JAMS</i> Monitor process on the local node or on all nodes in the VAXcluster. If you issue this command for the local node and the local node is running the Schedule process, the Schedule process will be stopped. However, if there are any nodes left in the VAXcluster which have a Monitor process running and are allowed to run the Schedule process, a new Schedule process will be started on one of the remaining nodes. If the JAMS_SCHEDULE process is running on the local node, it is also stopped. If this is a cluster-wide shutdown, the Monitor is stopped on all nodes in the cluster, and the Schedule process is stopped. During production use of <i>JAMS</i>, you should have the Monitor process running on all nodes which will run batch jobs.
-------------	---

restrictions	You must have the SYSPRV privilege to stop a JAMS Monitor process.
--------------	--

QUALIFIERS	/CLUSTER Directs the Monitor process to shutdown on all nodes in the VAXcluster. This implies that the Schedule process will also be shutdown. /NODE (default) Directs the Monitor process to shutdown on the local node. If the Schedule process is also running on this node, then it is also directed to shutdown.
------------	---

STOP NETWORK

STOP NETWORK

Stops the JAMS Network process.

FORMAT

STOP NETWORK

DESCRIPTION

This command stops the *JAMS* Network process and prevents the *JAMS* Monitor processes from starting a new one. Generally, this command is rarely used.

restrictions

You must have the SYSPRV privilege to stop the JAMS Network process.

STOP SCHEDULE

Stops the JAMS Schedule process.

FORMAT	STOP SCHEDULE
--------	---------------

DESCRIPTION	This command stops the <i>JAMS</i> Schedule process and prevents the <i>JAMS</i> Monitor processes from starting a new one. Generally, this command is used only to backup the <i>JAMS</i> database.
-------------	--

Caution: Jobs which are submitted by the *JAMS* Submit sub-system while the Schedule process is stopped will remain in a pending state until the Schedule process is restarted. Also, Jobs which are waiting on dependencies will remain in a pending state until the Schedule process is restarted and their dependencies are satisfied.

restrictions	You must have the SYSPRV privilege to stop the JAMS Schedule process.
--------------	---

QUALIFIERS	<i>/RESTART_DELAY=delay</i> <i>/NORESTART_DELAY</i> Specified, in minutes, a period of time after which the <i>JAMS</i> Schedule process will be automatically restarted. The default is <i>/RESTART_DELAY=60</i> .
------------	---

SUBMIT

SUBMIT

Submits a *JAMS* based batch job.

FORMAT **SUBMIT** *job-name-list*

PARAMETERS ***job-name-list***

The *job-name-list* parameter is the name of one or more Setup or Job definitions. If the *job-name-list* parameter contains more than one name, the names must be separated by commas.

You can submit a Job on a remote node by specifying the remote node name and access control information in the standard DECnet format.

DESCRIPTION

The *SUBMIT* command is very similar to the standard OpenVMS *SUBMIT* command. However, the *JAMS* version of the *SUBMIT* command is used to submit a *JAMS* Job or Setup definition, rather than a command file.

The *JAMS SUBMIT* command is used to submit one or more *JAMS* Jobs. You will be prompted for values for the Job's parameters and Report overrides. This prompting can be disabled with the */NOPROMPT* qualifier.

QUALIFIERS

OpenVMS SUBMIT Command Qualifiers

The following table lists the level of support provided by the *JAMS SUBMIT* command for the qualifiers supported by the standard OpenVMS *SUBMIT* command.

Qualifier	Support Provided
<i>/AFTER</i>	Supported
<i>/BACKUP</i>	NOT Supported
<i>/BEFORE</i>	NOT Supported
<i>/BY_OWNER</i>	NOT Supported
<i>/CHARACTERISTICS</i>	Supported
<i>/CLI</i>	Supported
<i>/CONFIRM</i>	NOT Supported
<i>/CPUTIME</i>	Supported
<i>/CREATED</i>	NOT Supported
<i>/DELETE</i>	Supported
<i>/EXCLUDE</i>	NOT Supported
<i>/EXPIRED</i>	NOT Supported

Qualifier	Support Provided
/HOLD	Supported
/IDENTIFY	Supported
/KEEP	Supported
/LOG_FILE	Supported
/MODIFIED	NOT Supported
/NAME	Supported
/NOTIFY	Supported
/PARAMETERS	Supported
/PRINTER	Supported
/PRIORITY	Supported
/QUEUE	Supported
/REMOTE	NOT Supported
/RESTART	Supported
/SINCE	NOT Supported
/USER	Supported
/WSDEFAULT	Supported
/WSEXTENT	Supported
/WSQUOTA	Supported

In addition to the supported OpenVMS SUBMIT command qualifiers, the following *JAMS* specific qualifiers are available:

/AGENT=agent-node

The /AGENT qualifier is used to specify the node name where this job's execution request should be sent. This node must be running the *JAMS* Agent or, if this is an rexec job, an rexec server.

/DEBUG

/NODEBUG (default)

The /DEBUG qualifier specifies that Jobs should be submitted in debug mode. This mode is used by programmers to test and debug Jobs. When a Job is submitted in debug mode, it is always submitted under the OpenVMS username of the person making the request. If the System definition specifies a OpenVMS username, it is ignored.

When submitting a Job in debug mode, the logical names used during Job parsing do not need to be defined at the executive level. This enables a programmer to define logical names in their process specific logical name tables which will direct the *JAMS* parsing routines to a test version of the Job source files and/or the Systems template library.

/DEPENDENCIES=(entry[,...])

The /DEPENDENCIES qualifier is used to force a Job to wait in a batch queue until the specified jobs have completed successfully. You specify the Jobs by their *JAMS* entry numbers. You can specify up to 32 entry numbers. If you specify only one entry number, you can omit the parentheses.

SUBMIT

This qualifier is very useful when you want to create a *stream* of Jobs. You can create one Job which submits all of the Jobs in the stream using this qualifier to define the stream dependencies.

/JOB

The */JOB* qualifier is used when you want to submit a Job definition without searching for a Setup definition first. If the */JOB* qualifier is present, a Job with the specified name will be searched for, and if it is found, submitted. Using this qualifier is the only way to submit a Job which has the same name as a Setup definition.

The default action, if neither */SETUP* or */JOB* are specified, is to search first for a Setup, then if the Setup was not found, search for a Job with the given name.

This is a positional qualifier. If it is supplied after the SUBMIT command, it applies to all of the names in the Job name list. If it is supplied after a name in the Job name list, it applies only to the preceding name.

/PROMPT (default)

/NOPROMPT

If you specify */NOPROMPT*, you will not be prompted for the values for a Job's parameters or Report overrides. Default values are taken instead.

/SET_SYMBOLS

/NOSET_SYMBOLS (default)

If you specify */SET_SYMBOLS*, a DCL symbol will be defined, or updated, for each of the Job's parameters. The symbol name will be the same as the parameter name, and the value will be the value used when submitting the Job.

If a symbol is being created, it is created in the local symbol table. If it is being updated, it is updated in whichever table it is currently defined.

/SETUP

The */SETUP* qualifier is used when you want to submit a Setup definition. If the */SETUP* qualifier is present, a Job with the specified name will *not* be searched for, and if the Setup does not exist, an error will be returned.

The default action, if neither */SETUP* or */JOB* are specified, is to search first for a Setup, then if the Setup was not found, search for a Job with the given name.

This is a positional qualifier. If it is supplied after the SUBMIT command, it applies to all of the names in the Job name list. If it is supplied after a name in the Job name list, it applies only to the preceding name.

/SUBMIT (default)

/NOSUBMIT

The */NOSUBMIT* qualifier is useful when you are testing a Job definition or a template library. If you specify */NOSUBMIT*, the job is not submitted to a batch queue. Instead, a file is created in the current default directory with the name of the Job and an extension of *.JAMS*. This file contains the parsed Job Source file exactly as it would be submitted to a batch queue.

/USE_SYMBOLS (default)***/NOUSE_SYMBOLS***

If the */USE_SYMBOLS* qualifiers is specified, either explicitly or by default, when *JAMS* is building a parameter screen, it will use the value of DCL symbols whose names match a parameter's name for the parameter's default value.

When *JAMS* determines the default value for a parameter, it takes the following steps:

- 1 Use the default value entered in the parameter definition.
- 2 If a Setup is being submitted, use the default value from the Setup's parameter screen.
- 3 If */USE_SYMBOLS* is specified, see if there is a DCL symbol with the same name as the parameter and use the symbol's value, if defined.
- 4 If a value was specified with the */PARAMETER* qualifier, use the specified value.

EXAMPLES

`JAMS> SUBMIT MYJOB`

This example searches for a Setup named MYJOB. If a Setup with this name is not found, it searches for a Job named MYJOB. If one of these is found, you are prompted for the Job's parameters, and then for the scheduled date and time and any report overrides.

`JAMS> SUBMIT APPLE:MYJOB`

This example searches the *JAMS* database on the remote node APPLE for a Setup named MYJOB. If a Setup with this name is not found, it searches for a Job named MYJOB. If one of these is found, you are prompted for the Job's parameters, and then for the scheduled date and time and any report overrides. The Job or Setup is submitted on node APPLE.

`JAMS> SUBMIT/NOSUBMIT TESTJOB`

This example does not actually submit a Job. It will go through the normal job submission sequence but, instead of submitting the job, it will create a file named TESTJOB.JAMS in the current default directory. This file contains the parsed and expanded command file of the job TESTJOB. The */NOSUBMIT* qualifier is used to test parameter substitution and template expansion.

3

JAMS Jobs

This chapter describes the batch jobs and utilities that are supplied with the *JAMS* system.

3.1 Report Headers

All of the reports produced by the *JAMS* System are prefaced with a *Report Header*. This header is a single page which supplies information on the Report and the Job which produced the Report. A sample of a Report Header is shown in Example 3–1, Report Header.

Example 3–1 Report Header

```
Job Access and Management System (JAMS)                Date: 01-31-91
Submit Recurring Jobs                                Time: 01:00 AM

      1
System I.D.: JAMS                Job Access & Management System (JAMS)

Setup Name: JAMS_AUTOSUBMIT 2
Description: Submit Recurring Jobs at their Scheduled Times

      Job Name: JAMS_AUTOSUBMIT 3
      Description: Submit Scheduled Jobs

      4                5
Submitted By: SYSTEM                on 29-JAN-1991 01:00

      6
Submit Jobs which are scheduled to run between 30-JAN-1991 06:00:00.00
                                         and 31-JAN-1991 06:00:00.00
```

- 1 Identification of the System to which the Job belongs.
 - 2 Identification of the Setup used to start this Job (if applicable).
 - 3 Identification of the Job which produced this Report.
 - 4 OpenVMS Username of the user who submitted this Job.
 - 5 Date and time the Job was submitted.
 - 6 Parameters supplied to the Job.
-

3.2 Jobs Supplied with JAMS

The following sections describe the Jobs which are supplied with the *JAMS* software. These Jobs are owned by the System “JAMS.”

JAMS_AUTOSUBMIT—Submit Recurring Jobs

The JAMS_AUTOSUBMIT job schedules Jobs based on Job and Setup definitions which specify automatic submission.

PARAMETERS

Submit Date

This date specifies the day for which you are scheduling Jobs. When JAMS_AUTOSUBMIT runs, it submits jobs which are scheduled to run in a 24 hour period. This period starts at the date and time specified by the parameters. All Jobs which are automatically submitted and are scheduled to run within this 24 hour period, will be submitted.

Submit Time

This time represents the starting time of day for scheduling Jobs. When this job runs, it submits jobs which are scheduled to run in a 24 hour period. This period starts at the date and time specified by the parameters.

Ignore the Last Auto-submit Date

This parameter should normally be “N”. The last thing that JAMS_AUTOSUBMIT does before submitting a Job or Setup, is to compare the date & time of the last time that this Job was automatically submitted to the current scheduled time. If the current scheduled date & time is on or before the last auto-submit date & time, JAMS_AUTOSUBMIT does *not* submit the Job. The Job or Setup appears in the schedule report with an error message.

This is done so that JAMS_AUTOSUBMIT can be rerun without double submitting Jobs. If you set this parameter to “Y”, JAMS_AUTOSUBMIT will not check the last auto-submit date & time.

DESCRIPTION

JAMS_AUTOSUBMIT schedules recurring jobs which are scheduled to run during a 24 hour period. This 24 hour period is defined by the parameters which are passed to the job.

This Job produces one report which is a detailed listing of the Jobs which were submitted. Example 3–2 is a sample of this report.

restrictions

This Job must run under a OpenVMS username which has, at a minimum, the following privileges:

- CMKRNL - So Jobs can be submitted under the username defined in the System definition.
- OPER - So that Jobs can be submitted to any queue.
- READALL - So that all Job command files can be read.
- SYSLOCK - To coordinate access to the monitor and schedule processes.

- SYSPRV - So the *JAMS* data files can be opened in update mode. This could also be done with ACL's.

Example 3-2 JAMS_AUTOSUBMIT Report

```

Submit Recurring Jobs at their Scheduled Times                                Page:          1
Jobs Submitted                                                                Date: 01-31-91
    1                                                                    Time: 01:00 AM
System I.D. JAMS                    Job Access & Management System (JAMS)
-----
Setup Name: JAMS_AUTOSUBMIT 2
Description: Submit Recurring Jobs at their Scheduled Times

Schedule For: DAILY 3
Except For:
Non-Workday: S (S = Schedule, D = Defer, I = Ignore)
Job Name: JAMS_AUTOSUBMIT
Description: Submit Scheduled Jobs
    4                                                                    5
Scheduled for: 31-JAN-1991 01:00:00.00 under Username: SYSTEM
Job JAMS_AUTOSUBMIT (queue SYS$BATCH, entry 334) pending 6
-----
Setup Name: JAMS_CLEANUP
Description: Clean up JAMS log and report file.

Schedule For: DAILY
Except For:
Non-Workday: S (S = Schedule, D = Defer, I = Ignore)
Job Name: JAMS_CLEANUP
Description: Cleanup JAMS log and report files.

Scheduled for: 31-JAN-1991 03:00:00.00 under Username: SYSTEM
Job JAMS_CLEANUP (queue SYS$BATCH, entry 335) holding until 31-JAN-1991 03:00
-----
    7
For System I.D. JAMS                    , Job Access & Management System (JAMS)
There were      2 jobs successfully submitted and      0 jobs with errors.

    8
    2 jobs were successfully submitted.
    0 jobs could not be submitted.

```

- 1 Identification of the System whose Jobs are being submitted.
 - 2 Setup Name.
 - 3 Scheduled Days.
 - 4 Actual Date and time the Job will be released.
 - 5 OpenVMS Username which the Job will run under.
 - 6 Reply text from submission to batch queue.
 - 7 Recap for indicated System.
 - 8 Recap for all Systems.
-

JAMS_AUTOSUBMIT_INTER.EXE—Submit Recurring Jobs Interactively

The JAMS_AUTOSUBMIT_INTER is an executable which does the same task as the JAMS_AUTOSUBMIT Job but is designed to be run interactively.

PARAMETERS

System I.D.

Specify a wildcard System I.D. and only Jobs and Setups whose System I.D. matches will be selected.

Job Name

Specify a wildcard Job Name and only Jobs that match this specification will be selected. Note that this selection applies only to individual Jobs, a Job which is part of a Setup is not filtered by this selection criteria.

Setup Name

Specify a wildcard Setup Name and only Setups which that this specification will be selected.

Start date/time

This date/time combination specifies the starting time of the range that jobs are being scheduled for.

End date/time

This date/time combination specifies the ending time of the range that jobs are being scheduled for.

Ignore the Last Auto-submit Date

This parameter should normally be “N”. The last thing that JAMS_AUTOSUBMIT does before submitting a Job or Setup, is to compare the date & time of the last time that this Job was automatically submitted to the current scheduled time. If the current scheduled date & time is on or before the last auto-submit date & time, JAMS_AUTOSUBMIT does *not* submit the Job.

This is done so that JAMS_AUTOSUBMIT(_INTER) can be rerun without double submitting Jobs. If you set this parameter to “Y”, JAMS_AUTOSUBMIT will not check the last auto-submit date & time.

If you are going to confirm each job the last autosubmit date processing is used to set the default value for the answer to the confirmation question which lets you override this setting by job.

Confirm Each Job

Specify if you want to review and confirm each job submitted to the schedule.

DESCRIPTION

The interactive version of JAMS_AUTOSUBMIT is useful when recovering from unusual events or after making major changes to your schedule.

Normally, the JAMS_AUTOSUBMIT Job runs once a day and submits 24 hours worth of jobs to the current schedule. If you are recovering from an unusual event you may want to use the JAMS_AUTOSUBMIT_INTER utility because you can process less time frames of less than 24 hours. You can also enter job selection criteria and manually confirm each selected job.

restrictions

This utility must run under a OpenVMS username which has, at a minimum, the following privileges:

- CMKRNL - So Jobs can be submitted under the username defined in the System definition.
- OPER - So that Jobs can be submitted to any queue.
- READALL - So that all Job command files can be read.
- SYSLCK - To coordinate access to the monitor and schedule processes.
- SYSPRV - So the *JAMS* data files can be opened in update mode. This could also be done with ACL's.

JAMS_CLEANUP—Cleanup Log and Report Files

The JAMS_CLEANUP job is responsible for cleaning up the Log and Report files generated by Jobs in the JAMS System.

PARAMETERS

Number of Log file versions to keep

This parameter specifies the minimum number of versions of a Job's log file which should be maintained.

Do not purge Logs created since

Log files which were created since this date will be retained even if it means that the number of versions retained will exceed the number specified in the versions parameter.

Delete all Logs created before

All versions of log files created before this date will be deleted.

DESCRIPTION

The JAMS_CLEANUP Job is used to purge and delete log and report files which are created by Jobs in the JAMS System.

Log files are purged to a maximum number of versions. However, if a log file was recently created, it will not be purged. The number of versions to retain and the minimum length of time to retain a log file are defined by the parameters.

Report files are deleted if the expiration date on the file has passed.

Note: This Job will cleanup the log and report files created by Jobs which are in the JAMS System. It is your responsibility to cleanup logs and reports created by jobs in Systems which you have defined.

JAMS_PURGE—Purge the JAMS History Files

The JAMS_PURGE job is responsible for purging old records from the History file.

PARAMETERS

Number of days of History to retain

This parameter specifies the maximum number of days which a Job execution history record should be retained.

Minimum Number of Records per Job

Specifies the minimum number of History records to keep per Job. This can prevent the purging of History records which are older than the specified maximum number of days. This is useful when a Job is run infrequently.

Maximum Number of Records per Job

Specifies the maximum number of History records to keep per Job. This will cause records to be purged even though they are not older than the specified maximum number of days. This is useful for Jobs which run many times a day.

DESCRIPTION

This job purges old records from the *JAMS* History file. The completion date of each history record is compared to the date defined by the parameter to this Job. If the History record is older than this date, it is eligible for deletion.

At least one history record for each Job is maintained, no matter how old it gets.

This Job does not produce any reports, but the Jobs log file will contain a message which tells you how many history records were deleted.

After the History file has been purged, the JAMS_SCHEDULE process is stopped and a *CONVERT/RECLAIM* is performed on the History file.

This job also creates a new JAMS.LOG file and purges JAMS.LOG to three versions. The JAMS.LOG file is used by the *JAMS* Monitor, Schedule and Network processes to log events.

JAMS1000—Print System Definitions

The JAMS1000 job prints a summary list of the System Definitions.

PARAMETERS *None.*

DESCRIPTION This Job prints a summary listing of the Systems defined in the JAMS database. Example 3–3 is a sample of this report.

Example 3–3 JAMS1000 Report

Job Access & Management System (JAMS)
Summary Listing of Systems

Page: 1
Date: 04-02-91
Time: 09:05 PM

System I.D.	Description	VMS Username
-----	-----	-----
AP	Accounts Payable	APBATCH
AR	Accounts Receivable	ARBATCH
BACKUP	Data Backup	BACKUP
GL	General Ledger	GLBATCH
INVOICE	Customer Invoice and Statement System	
INVTRY	Inventory Control	
JAMS	Job Access & Management System (JAMS)	
OPER	Operations	
PAYROLL	Employee Compensation & Benefits	PAYBATCH
SYSTEM	System Management and Control	
TEST	Test System	

11 Systems Printed

JAMS1001—Print System Definitions

The JAMS1001 job prints a detailed list of the System Definitions.

PARAMETERS *System I.D.*

The System I.D. parameter is used to specify a single System I.D. to print. If you do not supply a value for this parameter, all of the Systems are printed.

DESCRIPTION This Job prints a detailed listing of one or more Systems defined in the JAMS database. Example 3–4 is a sample of this report.

Example 3–4 JAMS1001 Report

```

Job Access & Management System (JAMS)                               Page:          1
Detailed List of Systems                                           Date: 04-02-91
                                                                    Time: 09:05 PM

        System I.D.: AP
        Description: Accounts Payable

        Default Batch Queue:
        Default Print Queue:

        Log File Print Queue:
        Log File Directory:
        Print Log Files: Y
        Keep Log Files: N

        .JOB/.COM File Directory:
        Print File Directory:
        JAMS Template Library:

        VMS Username: APBATCH
        Priority Modifier:      0

Send notification of abnormal Job termination to:
        VMS Mail Address List: FRANK,JOEW
        Broadcast Username List: FRANK
        Operator Classs List: CENTRAL

        Access Control List-----
        (IDENTIFIER= [MVP,JOHN] ,ACCESS=CONTROL+SUBMIT+DEBUG+JOB_ADD+
        JOB_CHANGE+JOB_INQUIRE+JOB_DELETE+DEFINE_SETUP)
        End of ACL-----

```

JAMS1100—Print Job Definitions

The JAMS1100 job prints a summary list of Job Definitions.

PARAMETERS

System I.D.

Only Jobs in the System specified by this parameter are printed. If you leave this parameter blank, Jobs in all Systems will be printed.

DESCRIPTION

This Job prints a summary listing of Job definitions. Example 3–5 is a sample of this report.

Example 3–5 JAMS1100 Report

Job Access & Management System (JAMS)
Summary Listing of Jobs, by System I.D.

Page: 1
Date: 04-02-91
Time: 09:05 PM

Jobs in System: AP	Accounts Payable
Job Name	Description

JAP100	Invoice Edit Listing
JAP110	Post Invoices
JAP120	Cash Requirements
JAP130	Print Checks
JAP140	Update A/P History

5 Jobs in this System.

.
.
.

Job Access & Management System (JAMS)
Summary Listing of Jobs, by System I.D.

Page: 2
Date: 04-02-91
Time: 09:05 PM

Jobs in System: AP	Accounts Payable
Job Name	Description

FULL_BACKUP	Full Backup of all Disks
IMAGE_BACKUP	Image Backup of Specified Disk

2 Jobs in this System.

JAMS1101—Print Job Definitions

The JAMS1101 job prints a detailed list of Job Definitions.

PARAMETERS

System I.D.

Only Jobs in the System specified by this parameter are printed. If you leave this parameter blank, Jobs in all Systems will be printed.

Job Name

This parameter is used to select Jobs which should be printed. You can use standard VMS wildcard characters in the selection string. Only Jobs which match the string specified and match the System I.D. supplied will be printed.

Include Reports

You have the option of including a Jobs Report definitions in the report generated by this Job. If you specify "Y", Report definitions will be listed.

Include Parameters

You have the option of including a Jobs Parameter definitions in the report generated by this Job. If you specify "Y", Parameter definitions will be listed.

DESCRIPTION

This Job prints a detailed listing of Job definitions. Example 3–6 is a sample of this report.

JAMS1101

Example 3-6 JAMS1101 Report

Job Access & Management System (JAMS)
Detail Job Definitions

Page: 1
Date: 04-02-91
Time: 09:05 PM

Jobs in System: PAYROLL Employee Compensation & Benefits

Job Name	Description
JPAY100	Time Card Edit
Source File	
Override Batch Queue	
Scheduling Priority Modifier	0
Restartable	N
Parse	N
Job may be submitted between	: and :
Report Name	TIME_EDIT
Description	Time Card Edit Report
Logical Name	
Override Print Queue	
Print Form	
Number of Copies	1
Retention Days	0
Print Qualifiers	
JPAY110	Gross to Net Calculation
Source File	
Override Batch Queue	
Scheduling Priority Modifier	0
Restartable	N
Parse	N
Job may be submitted between	: and :
Report Name	GROSS_NET
Description	Gross to Net
Logical Name	
Override Print Queue	
Print Form	
Number of Copies	1
Retention Days	0
Print Qualifiers	

JAMS1200—Print Setup Definitions

The JAMS1200 job prints a summary list of Setup Definition by Job.

PARAMETERS *System I.D.*

Only Setups for Jobs in the System specified by this parameter are printed. If you leave this parameter blank, Setups in all Systems will be printed.

DESCRIPTION

This Job prints a summary listing of Setup definitions by Job. Example 3-7 is a sample of this report.

Example 3-7 JAMS1200 Report

Job Access & Management System (JAMS)
Summary Listing of Setups by Job

Page: 1
Date: 03-21-93
Time: 04:21 PM

System: TEST Test System

Job Name: JAP100
 Invoice Edit Listing

Setup Name: JAP100
 Invoice Edit Listing

Schedule for: DAILY
Except for: TUE,THR
 Time: 08:30 PM

Job Name: JAP110
 Post Invoices

Setup Name: JAP110
 Post Invoices

Schedule for: DAILY
Except for:
 Time: 07:00 PM

JAMS2000—Print Summary Schedule

The JAMS2000 job generates and prints a summary of the Jobs which should run between any two points in time.

PARAMETERS

System I.D.

Only Jobs in the System specified by this parameter are printed. If you leave this parameter blank, Jobs in all Systems will be printed.

Start Date/Time

These two parameters specify the starting date and time of the period for which a schedule should be generated.

End Date/Time

These two parameters specify the ending date and time of the period for which a schedule should be generated.

DESCRIPTION

This Job generates a report of the Jobs which should run between any two points in time. This report is for informational purposes only. The Jobs on this report will *not* be submitted.

This report is produced by examining the Setup and Job definitions. All Setups and Jobs which specify a Scheduled Date which would fall into the specified date range will be included on this report. The report is *not* limited to recurring Jobs.

Example 3–8 is a sample of this report.

Example 3-8 JAMS2000 Report

Job Access & Management System (JAMS)
Scheduled Jobs Between Two Dates

Page: 1
Date: 04-02-91
Time: 09:06 PM

System I.D. Description

AP Accounts Payable

Setup Name: JAP130 **1**
Description: Print Checks

Schedule For: THURSDAY **2**
Non-Workday processing D (S = Schedule, D = Defer, I = Ignore)

Scheduled to run at 4-APR-1991 15:00:00.00 **3**
Scheduled to run at 11-APR-1991 15:00:00.00

Setup Name: JAP120
Description: Cash Requirements

Schedule For: THURSDAY
Non-Workday processing D (S = Schedule, D = Defer, I = Ignore)

Scheduled to run at 4-APR-1991 10:00:00.00
Scheduled to run at 11-APR-1991 10:00:00.00

Setup Name: JAP110
Description: Post Invoices

Schedule For: DAILY
Non-Workday processing I (S = Schedule, D = Defer, I = Ignore)

Scheduled to run at 1-APR-1991 19:00:00.00
Scheduled to run at 2-APR-1991 19:00:00.00
Scheduled to run at 3-APR-1991 19:00:00.00
Scheduled to run at 4-APR-1991 19:00:00.00
Scheduled to run at 5-APR-1991 19:00:00.00 **4**
Scheduled to run at 8-APR-1991 19:00:00.00
Scheduled to run at 9-APR-1991 19:00:00.00
Scheduled to run at 10-APR-1991 19:00:00.00
Scheduled to run at 11-APR-1991 19:00:00.00
Scheduled to run at 12-APR-1991 19:00:00.00

- 1** The name and description of the Setup definition which is scheduled to run during the requested period of time.
- 2** The schedule definition.
- 3** A list of the dates and times on which this job should run.
- 4** This Setup is scheduled to run daily but is ignored on non-workdays. Note that the 6th and 7th (Saturday & Sunday) are skipped.

JAMS2100—Print Simulated Schedule

The JAMS2100 job generates and prints a summary of the Jobs which should run between any two points in time. It also simulates the execution of the scheduled jobs to determine which Triggers may fire and how jobs may be affected by their dependencies.

PARAMETERS

System I.D.

Only Jobs in the System specified by this parameter are printed. If you leave this parameter blank, Jobs in all Systems will be printed.

Start Date/Time

These two parameters specify the starting date and time of the period for which a schedule should be generated.

End Date/Time

These two parameters specify the ending date and time of the period for which a schedule should be generated.

Include Planned Jobs

If true, Jobs and Setups which are not automatically submitted will also be included in the simulation if their schedule criteria falls into the simulation date/time range.

Include Repeated Jobs

If true, all occurrences of a Job or Setup which periodically repeats will be included in the simulation.

Include Triggered Jobs

If true, Jobs which would be submitted because of a Trigger firing will be included in the simulation.

DESCRIPTION

This Job generates a report of the Jobs which should run between any two points in time. This report is for informational purposes only. The Jobs on this report will *not* be submitted.

This report is produced by examining the Setup and Job definitions. All Setups and Jobs which specify a Scheduled Date which would fall into the specified date range will be included on this report. The report is *not* limited to recurring Jobs.

JAMS3000—Print Job Execution History

The JAMS3000 job produces a report of the Jobs which executed during a specified period of time.

PARAMETERS

System I.D.

Only Jobs in the System specified by this parameter are printed. If you leave this parameter blank, Jobs in all Systems will be printed.

Job Name

This parameter is used to select Jobs which should be printed. You can use standard VMS wildcard characters in the selection string. Only Jobs which match the string specified and match the System I.D. supplied, will be printed.

Setup Name

This parameter is used to select Jobs which should be printed. You can use standard VMS wildcard characters in the selection string. Only Jobs which were submitted via the supplied Setup and match the System I.D. supplied, will be printed.

Start Date/Time

These two parameters specify the starting date and time of the period for which execution history should be printed.

End Date/Time

These two parameters specify the ending date and time of the period for which execution history should be printed.

Include Severities

These parameters can be used to select Jobs based upon the severity of the Job's final status.

DESCRIPTION

This Job produces a report of the Jobs which executed during a specified period of time. This report can be used as an audit trail and/or archive report.

Example 3–9 is a sample of this report.

JAMS Jobs

JAMS3000

Example 3-9 JAMS3000 Report

Job Access & Management System (JAMS)
Job Execution History

Page: 1
Date: 04-02-91
Time: 09:06 PM

System I.D. Description

JAMS Job Access & Management System (JAMS)

Job Name: JAMS_AUTOSUBMIT
Description: Submit Scheduled Jobs

Submitted	Started	Completed	Status
1-APR-1991 17:51:34.02	1-APR-1991 17:52:19.24	1-APR-1991 17:52:51.75	Success
2-APR-1991 01:00:08.66			Info.

1

Job Name: JAMS_CLEANUP
Description: Cleanup JAMS log and report files.

Submitted	Started	Completed	Status
1-APR-1991 17:51:37.80	1-APR-1991 17:52:52.28	1-APR-1991 17:53:00.61	Success
2-APR-1991 01:00:10.90			Info.

For System I.D. JAMS Job Access & Management System (JAMS)
There were 11 Jobs which ran between 1-APR-1991 00:00:00.00
and 2-APR-1991 00:00:00.00

1 This Job was submitted but has not started.

4

Using the JAMS Callable Interface

This chapter explains how to call *JAMS* from a user written program. It also explains how you can have *JAMS* call your routines when certain events occurs.

4.1 Using The Callable Interface

The *JAMS* callable interface is provided so that you can integrate batch jobs into your application software. You can call *JAMS* routines which will submit a single batch job or display a menu of jobs and allow the end-user to select a job.

All of the capabilities of the *JAMS* submission interface are available from the callable interface. The person selecting by way of the callable interface will be prompted for parameters and report overrides in the same manner as a user of the *JAMS_MASTER* program.

4.1.1 Calling JAMS Routines

The *JAMS* callable interface conforms to the OpenVMS Procedure Calling and Condition Handling Standard. This conformance allows the *JAMS* routines to be called from any programming language which supports the standard. All of the Compaq supplied compilers support the OpenVMS Procedure Calling and Condition Handling Standard.

4.1.2 Using Item Lists

Because of the extensive and diverse nature of the arguments used by the *JAMS* routines, many arguments are passed using an *item list*.

An item list is an array of *item descriptors*. This array is defined and initialized in your application program and then the address of the item list is passed to the *JAMS* routine. An item descriptor consists of four fields, two fields are words (16 bits each) and two fields are longwords (32 bits each). The item list array is terminated by an item descriptor with zero in all fields.

The structure of an item list is the same for all programming languages, but the methods used to define an item list are different in every language. Please refer to your language specific documentation for information on how to define data structures such as these. The fields in an item descriptor are shown in the following table:

Using the JAMS Callable Interface

Using The Callable Interface

Field	Size	Function
Buffer Length	16 bits	Specifies the length of input items and the maximum length of output items.
Item Code	16 bits	An identifier. It specifies what this item descriptor is providing or requesting.
Buffer Address	32 bits	Specifies the address of the input or output buffer. Some input item codes may use this field directly.
Return Length Address	32 bits	For output item codes, this is the address of a longword where the actual length of the returned data is stored. For input item codes, this field is not used and should be set to zero.

4.1.3 Linking with JAMSSHR.EXE

The *JAMS* callable interface is provided in the form of a sharable image. Your program must be linked against this sharable image in order to resolve references to *JAMS* symbols and routines. Since the *JAMS* interface is provided as a sharable image, when you install new versions of *JAMS*, you will *not* need to recompile or re-link your programs. However, the *JAMS* sharable image is not inserted into the default OpenVMS sharable image library so you must explicitly specify the *JAMS* sharable image when you link your programs. The following example shows how this is done:

```
$ COBOL myprogram.cob
$ LINK myprogram,SYS$INPUT/OPTION
JAMSSHR/SHARE
Ctrl/Z
$
```

4.1.4 Privileges

You do not need to have any special privileges to compile and link programs which call *JAMS* routines. You will need special privileges to execute the program. In order to execute a program which calls *JAMS* routines, the process which runs the program needs the privileges listed in the following table. You could also install your program with the required privileges.

Privilege	Reason
SYSPRV	Needed to gain write access to the JAMS data files. This could also be accomplished with ACL's.
SYSLCK	Required to coordinate access to the monitor and schedule processes.
OPER	May be required to submit jobs to the proper queue.
CMKRNL	Required if your program will be submitting Jobs which run under a different OpenVMS username.

4.2 Sample programs

JAMS provides a number of sample programs which are located in *JAMS_EXAMPLES*:

JAMS2100.COB

Illustrates the use of the *JAMS_SCHEDULE_xxx* routines. This prints a projected schedule.

JAMS3000.COB

Illustrates the use of the *JAMS_HISTORY_xxx* routines. This prints Job execution history.

JAMS_EXAMPLE.COB

This program is written in OpenVMS COBOL and demonstrates how to call the *JAMS_SUBMIT_JOB* routine.

JAMS_SCANLOG.C

This program is written in C and demonstrates how to write a *JAMS_SCHCALLUSER* routine. This routine is called by the *JAMS_SCHEDULE* process when events occur. The *JAMS_SCANLOG* is an example of scanning a Jobs .LOG file to determine if it was successful.

JAMS_SCHCALLUSER.COB

This program is written in OpenVMS COBOL and demonstrates how to write a *JAMS_SCHCALLUSER* routine. This routine is called by the *JAMS_SCHEDULE* process when events occur.

JAMS_DECALERT.C

This program is written in OpenVMS C and demonstrates how you could activate DECalert from a *JAMS_SCHCALLUSER* routine.

4.3 Routines

The following sections describe the routines available to your application programs. The format of the *JAMS_SCHCALLUSER* routine is also explained.

JAMS_EVALUATE_DATE—Evaluate an English language date specification.

The JAMS_EVALUATE_DATE routine is used to parse an English language date specification. JAMS_EVALUATE_DATE.

FORMAT	JAMS_EVALUATE_DATE	<i>input-string, output-date, [todays-date]</i>
---------------	---------------------------	---

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
----------------	---

ARGUMENTS	<i>input-string</i> VMS Usage: string type: string access: read only mechanism: by descriptor The date specification to be evaluated.
------------------	---

	<i>output-date</i> VMS Usage: VMS quadword Date type: quadword access: write only mechanism: by reference The resulting date.
--	---

	<i>todays-date</i> VMS Usage: VMS quadword Date type: quadword access: read only mechanism: by reference The date which should be considered today's date while evaluating the date specification.
--	--

DESCRIPTION	The <i>JAMS_EVALUATE_DATE</i> routine can be called when you want to convert an English language date specification into a specific date.
--------------------	---

**RETURN
VALUES**

INVPERIOD	Invalid period.
INVSCHDATE	Invalid Schedule Date.
DATESNSU	Dates for this period have not been defined.
JAMS_INVRETDSC	Invalid return descriptor.

JAMS_GET_DATA—Return data from a passed context

The JAMS_GET_DATA routine is used to obtain data associated with a context.

FORMAT	JAMS_GET_DATA <i>context,</i> <i>data-ident1, return-desc1,</i> <i>[data-ident2, return-desc2]...</i>
---------------	--

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
----------------	---

ARGUMENTS	<i>context</i> VMS Usage: context type: longword (unsigned) access: read only mechanism: by value This is a context variable which was passed, or returned to your routine by JAMS. The JAMS_GET_DATA routine can be called only with a context which has been: • passed to your code via a JAMS calluser routine, • returned to your code via a call to JAMS_SCHEDULE_GET_JOB, • returned to your code via a call to JAMS_SCHEDULE_GET_DEPEND. • returned to your code via a call to JAMS_HISTORY_GET.
------------------	--

data-identN

VMS Usage: **longword**
type: **longword**
access: **read only**
mechanism: **by value**

The *data-ident* parameter identifies a data element which you want JAMS_GET_DATA to return to you.

return-descN

VMS Usage: **various**
type: **various**
access: **write only**
mechanism: **by descriptor**

The *return-desc* parameter is a descriptor which points to the location of the variable which is to receive the requested data element.

DESCRIPTION

When calling *JAMS_GET_DATA* you pass one or more pairs of *data-ident* and *return-desc* parameters.

The data identifiers supported vary based on the type of context. *JAMS* can call your code with the following context types:

JAMS_K_SCHSTART	JAMS_SCHEDULE process is starting up.
JAMS_K_SCHSTOP	JAMS_SCHEDULE process is stopping.
JAMS_K_JOBSTART	A Job is starting to execute.
JAMS_K_JOBEND	A Job has completed.
n/a	A context returned from JAMS_SCHEDULE_GET_JOB.
n/a	A context returned from JAMS_SCHEDULE_GET_DEPEND.
n/a	A context returned from JAMS_HISTORY_GET.

JAMS_K_JOBSTART Context

The following data identifiers are supported by the *JAMS_GET_DATA* routine when the context type is *JAMS_K_JOBSTART*:

JAMS_K_CHECKPOINT_DATA
 JAMS_K_CPU_RATING
 JAMS_K_DESC_LN1
 JAMS_K_DESC_LN2
 JAMS_K_ENTRY
 JAMS_K_FINAL_STATUS
 JAMS_K_JOB_DESC_LN1
 JAMS_K_JOB_DESC_LN2
 JAMS_K_JOB_NAME
 JAMS_K_LOG_FILENAME
 JAMS_K_MAIL_ADR
 JAMS_K_NAME
 JAMS_K_NODENAME
 JAMS_K_OPER_CLASSES
 JAMS_K_PID
 JAMS_K_PRINT_DIR
 JAMS_K_QUEUE_NAME
 JAMS_K_REPLY_USERS
 JAMS_K_RESTART_COUNT
 JAMS_K_RON
 JAMS_K_RUN_UNDER
 JAMS_K_SETUP_DESC_LN1
 JAMS_K_SETUP_DESC_LN2
 JAMS_K_SETUP_NAME
 JAMS_K_START_TIME
 JAMS_K_SUBMITTED_BY
 JAMS_K_SUBMIT_TIME
 JAMS_K_SYSTEM_DESC
 JAMS_K_SYSTEM_ID

JAMS_GET_DATA

JAMS_K_JOBEND Context

The following data identifiers are supported by the *JAMS_GET_DATA* routine when the context type is *JAMS_K_JOBEND*:

JAMS_K_BIOCNT
JAMS_K_CHECKPOINT_DATA
JAMS_K_CPUTIM
JAMS_K_CPU_RATING
JAMS_K_DESC_LN1
JAMS_K_DESC_LN2
JAMS_K_DIOCNT
JAMS_K_END_TIME
JAMS_K_ENTRY
JAMS_K_FINAL_STATUS
JAMS_K_JOB_DESC_LN1
JAMS_K_JOB_DESC_LN2
JAMS_K_JOB_NAME
JAMS_K_LOG_FILENAME
JAMS_K_MAIL_ADR
JAMS_K_NAME
JAMS_K_NODENAME
JAMS_K_OPER_CLASSES
JAMS_K_PAGEFLTS
JAMS_K_PGFLPEAK
JAMS_K_PID
JAMS_K_PRINT_DIR
JAMS_K_QUEUE_NAME
JAMS_K_REPLY_USERS
JAMS_K_RESTART_COUNT
JAMS_K_ROM
JAMS_K_RUN_UNDER
JAMS_K_SETUP_DESC_LN1
JAMS_K_SETUP_DESC_LN2
JAMS_K_SETUP_NAME
JAMS_K_START_TIME
JAMS_K_SUBMITTED_BY
JAMS_K_SUBMIT_TIME
JAMS_K_SYSTEM_DESC
JAMS_K_SYSTEM_ID
JAMS_K_VOLUMES
JAMS_K_WSPEAK

JAMS_SCHEDULE_GET_JOB Context

The following data identifiers are supported by the *JAMS_GET_DATA* routine when the context was returned by a call to *JAMS_SCHEDULE_GET_JOB*:

JAMS_K_AUTOSUBMIT
JAMS_K_AVG_BIOCNT
JAMS_K_AVG_CPUTIM
JAMS_K_AVG_DIOCNT
JAMS_K_AVG_PAGEFLTS
JAMS_K_AVG_PGFLPEAK

JAMS_K_AVG_WSPEAK
JAMS_K_DESC_LN1
JAMS_K_DESC_LN2
JAMS_K_END_TIME
JAMS_K_ESTIMATED_TIME
JAMS_K_JOB_DESC_LN1
JAMS_K_JOB_DESC_LN2
JAMS_K_JOB_NAME
JAMS_K_LAST_AUTOSUBMIT
JAMS_K_LAST_ERROR
JAMS_K_LAST_FATAL
JAMS_K_LAST_INFO
JAMS_K_LAST_SUCCESS
JAMS_K_LAST_WARNING
JAMS_K_MAIL_ADR
JAMS_K_MAX_BIOCNT
JAMS_K_MAX_CPUTIM
JAMS_K_MAX_DIOCNT
JAMS_K_MAX_PAGEFLTS
JAMS_K_MAX_PGFLPEAK
JAMS_K_MAX_WSPEAK
JAMS_K_MIN_BIOCNT
JAMS_K_MIN_CPUTIM
JAMS_K_MIN_DIOCNT
JAMS_K_MIN_PAGEFLTS
JAMS_K_MIN_PGFLPEAK
JAMS_K_MIN_WSPEAK
JAMS_K_NAME
JAMS_K_NONWORKDAY_OPTION
JAMS_K_NOTIFY_SEVERITY
JAMS_K_OPER_CLASSES
JAMS_K_RECOVERY_SEVERITY
JAMS_K_REPLY_USERS
JAMS_K_RESTARTABLE
JAMS_K_RESUBMIT_BASE
JAMS_K_RESUBMIT_DELAY
JAMS_K_RESUBMIT_END
JAMS_K_SCHEDULE_TYPE
JAMS_K_SCHED_DATE
JAMS_K_SCHED_EXCEPT
JAMS_K_SCHED_TIME
JAMS_K_SETUP_DESC_LN1
JAMS_K_SETUP_DESC_LN2
JAMS_K_SETUP_NAME
JAMS_K_SUBMIT_FROM
JAMS_K_SUBMIT_HOLD
JAMS_K_SUBMIT_TO
JAMS_K_SYSTEM_DESC
JAMS_K_SYSTEM_ID

JAMS_GET_DATA

JAMS_SCHEDULE_GET_DEPEND Context

The following data identifiers are supported by the *JAMS_GET_DATA* routine when the context was returned by a call to *JAMS_SCHEDULE_GET_DEPEND*:

JAMS_K_AVG_BIOCNT
JAMS_K_AVG_CPUTIM
JAMS_K_AVG_DIOCNT
JAMS_K_AVG_PAGEFLTS
JAMS_K_AVG_PGFLPEAK
JAMS_K_AVG_WSPEAK
JAMS_K_DEPENDENCY
JAMS_K_DESC_LN1
JAMS_K_DESC_LN2
JAMS_K_JOB_DESC_LN1
JAMS_K_JOB_DESC_LN2
JAMS_K_JOB_NAME
JAMS_K_LAST_ERROR
JAMS_K_LAST_FATAL
JAMS_K_LAST_INFO
JAMS_K_LAST_SUCCESS
JAMS_K_LAST_WARNING
JAMS_K_MAX_BIOCNT
JAMS_K_MAX_CPUTIM
JAMS_K_MAX_DIOCNT
JAMS_K_MAX_PAGEFLTS
JAMS_K_MAX_PGFLPEAK
JAMS_K_MAX_WSPEAK
JAMS_K_MIN_BIOCNT
JAMS_K_MIN_CPUTIM
JAMS_K_MIN_DIOCNT
JAMS_K_MIN_PAGEFLTS
JAMS_K_MIN_PGFLPEAK
JAMS_K_MIN_WSPEAK
JAMS_K_NAME
JAMS_K_NOTIFY_SEVERITY
JAMS_K_RECOVERY_SEVERITY
JAMS_K_RESTARTABLE
JAMS_K_SATISFIED
JAMS_K_SATISFIED_TIME
JAMS_K_SINCE_DESC_LN1
JAMS_K_SINCE_DESC_LN2
JAMS_K_SINCE_NAME
JAMS_K_SINCE_TIME
JAMS_K_SUBMIT_FROM
JAMS_K_SUBMIT_HOLD
JAMS_K_SUBMIT_TO
JAMS_K_SYSTEM_ID

JAMS_HISTORY_GET Context

The following data identifiers are supported by the *JAMS_GET_DATA* routine when the context was returned by a call to *JAMS_HISTORY_GET*:

JAMS_K_AUTOSUBMIT
 JAMS_K_AVG_BIOCNT
 JAMS_K_AVG_CPUTIM
 JAMS_K_AVG_DIOCNT
 JAMS_K_AVG_PAGEFLTS
 JAMS_K_AVG_PGFLPEAK
 JAMS_K_AVG_WSPEAK
 JAMS_K_BIOCNT
 JAMS_K_CHECKPOINT_DATA
 JAMS_K_CPUTIM
 JAMS_K_CPU_RATING
 JAMS_K_DESC_LN1
 JAMS_K_DESC_LN2
 JAMS_K_DIOCNT
 JAMS_K_END_TIME
 JAMS_K_ENTRY
 JAMS_K_FINAL_STATUS
 JAMS_K_JOB_DESC_LN1
 JAMS_K_JOB_DESC_LN2
 JAMS_K_JOB_NAME
 JAMS_K_LAST_AUTOSUBMIT
 JAMS_K_LAST_ERROR
 JAMS_K_LAST_FATAL
 JAMS_K_LAST_INFO
 JAMS_K_LAST_SUCCESS
 JAMS_K_LAST_WARNING
 JAMS_K_LOG_FILENAME
 JAMS_K_MAIL_ADR
 JAMS_K_MAX_BIOCNT
 JAMS_K_MAX_CPUTIM
 JAMS_K_MAX_DIOCNT
 JAMS_K_MAX_PAGEFLTS
 JAMS_K_MAX_PGFLPEAK
 JAMS_K_MAX_WSPEAK
 JAMS_K_MIN_BIOCNT
 JAMS_K_MIN_CPUTIM
 JAMS_K_MIN_DIOCNT
 JAMS_K_MIN_PAGEFLTS
 JAMS_K_MIN_PGFLPEAK
 JAMS_K_MIN_WSPEAK
 JAMS_K_NAME
 JAMS_K_NODENAME
 JAMS_K_NONWORKDAY_OPTION
 JAMS_K_NOTIFY_SEVERITY
 JAMS_K_OPER_CLASSES
 JAMS_K_OVERRIDE_NAME
 JAMS_K_PAGEFLTS
 JAMS_K_PGFLPEAK
 JAMS_K_PID

JAMS_GET_DATA

JAMS_K_QUEUE_NAME
JAMS_K_RECOVERY_SEVERITY
JAMS_K_REPLY_USERS
JAMS_K_RESTARTABLE
JAMS_K_RESTART_COUNT
JAMS_K_RESTART_NUMBER
JAMS_K_RESUBMIT_BASE
JAMS_K_RESUBMIT_DELAY
JAMS_K_RESUBMIT_END
JAMS_K_ROM
JAMS_K_RUN_UNDER
JAMS_K_SCHED_DATE
JAMS_K_SCHED_EXCEPT
JAMS_K_SETUP_DESC_LN1
JAMS_K_SETUP_DESC_LN2
JAMS_K_SETUP_NAME
JAMS_K_START_TIME
JAMS_K_SUBMITTED_BY
JAMS_K_SUBMIT_FROM
JAMS_K_SUBMIT_HOLD
JAMS_K_SUBMIT_TIME
JAMS_K_SUBMIT_TO
JAMS_K_SYSTEM_DESC
JAMS_K_SYSTEM_ID
JAMS_K_WSPEAK

Descriptions of Item Codes

JAMS_K_AUTOSUBMIT

Returns the Autosubmit flag of a Setup definition. This flag will be “Y” or “N”. The return descriptor must point to a character string variable.

JAMS_K_AVG_BIOCNT

Returns the average number of buffered I/O's performed by the job. The return descriptor must point to an integer variable.

JAMS_K_AVG_CPUTIM

Returns the average amount of CPU time, in 10 millisecond units, used by the job. The return descriptor must point to an integer variable.

JAMS_K_AVG_DIOCNT

Returns the average number of direct I/Os performed by the job. The return descriptor must point to an integer variable.

JAMS_K_AVG_PAGEFLTS

Returns the average number of page faults which incurred by the job. The return descriptor must point to an integer variable.

JAMS_K_AVG_PGFLPEAK

Returns the average peak page file usage of the job. The return descriptor must point to an integer variable.

JAMS_K_AVG_WSPEAK

Returns the average peak working set of the job. The return descriptor must point to an integer variable.

JAMS_K_BIOCNT

Returns the number of buffered I/O's performed by the job. The return descriptor must point to an integer variable.

JAMS_K_CHECKPOINT_DATA

Returns the last value of the job's checkpoint data. Only the first 39 bytes of the checkpoint data can be returned. The return descriptor must point to a character string variable.

JAMS_K_CPUTIM

Returns the amount of CPU time, in 10 millisecond units, used by the job. The return descriptor must point to an integer variable.

JAMS_K_CPU_RATING

Returns the value of the JAMS_CPU_RATING logical name for the node on which the job executed. The return descriptor must point to a F-float floating point variable.

JAMS_K_DEPENDENCY

Returns a text representation, or description, of the the dependency. The return descriptor must point to a character string variable.

JAMS_K_DESC_LN1, JAMS_K_DESC_LN2

Returns line one or line two of the description of the job. If the job was submitted via a Setup definition, the description is from the Setup definition otherwise the description is from the Job definition. If the context is from a call to *JAMS_SCHEDULE_GET_DEPEND* then this item code returns line one or line two of the description of the Job or Variable involved in the dependency. The description may be up to 50 characters in length. The return descriptor must point to a character string variable.

JAMS_K_DIOCNT

Returns the number of direct I/Os performed by the job. The return descriptor must point to an integer variable.

JAMS_GET_DATA

JAMS_K_END_TIME

Returns the actual or estimated completion time of the job as a VMS quadword date and time. If the context is from a call to *JAMS_SCHEDULE_GET_JOB* then this time is estimated based upon the estimated start time and the Job's (or Setup's) average elapsed time. The return descriptor must point to an 8 byte variable.

JAMS_K_ENTRY

Returns the VMS queue entry number of the job. The return descriptor must point to an integer variable.

JAMS_K_ESTIMATED_TIME

Returns the estimated start time of the job as a VMS quadword date and time. This estimate is based upon the scheduled time and the time when all dependencies are satisfied. The return descriptor must point to an 8 byte variable.

JAMS_K_FINAL_STATUS

Returns the final status of the job. The return descriptor must point to an integer variable.

JAMS_K_JOB_DESC_LN1, JAMS_K_JOB_DESC_LN2

Returns line one or line two of the description of the job. This description comes from the Job definition. The description may be up to 50 characters in length. The return descriptor must point to a character string variable.

JAMS_K_JOB_NAME

Returns the name of the job as specified in the Job Definition. A job name may be up to 39 characters in length. The return descriptor must point to a character string variable.

JAMS_K_LAST_AUTOSUBMIT

Returns the date and time of the most recent automatic submit of this Setup. This date is updated only when a Setup is automatically submitted by the JAMS_AUTOSUBMIT Job. The return descriptor must point to an 8 byte variable.

JAMS_K_LAST_ERROR

Returns the date and time of the most recent error completion of this Job or Setup. The return descriptor must point to an 8 byte variable.

JAMS_K_LAST_FATAL

Returns the date and time of the most recent fatal completion of this Job or Setup. The return descriptor must point to an 8 byte variable.

JAMS_K_LAST_INFO

Returns the date and time of the most recent informational completion of this Job or Setup. The return descriptor must point to an 8 byte variable.

JAMS_K_LAST_SUCCESS

Returns the date and time of the most recent successful completion of this Job or Setup. The return descriptor must point to an 8 byte variable.

JAMS_K_LAST_WARNING

Returns the date and time of the most recent warning completion of this Job or Setup. The return descriptor must point to an 8 byte variable.

JAMS_K_LOG_FILENAME

Returns the full file specification, including version number, of the Job's .LOG file. The log file name is available only if the Job actually started and got as far as running JAMS_REGISTRAR.EXE in the SYLOGIN command file.

JAMS_K_MAIL_ADR

Returns the VMS mail addresses specified in the job's System I.D. or, if the job is unregistered, in the *JAMS* configuration. The list of addresses may be up to 256 characters in length. The return descriptor must point to a character string variable.

JAMS_K_MAX_BIOCNT

Returns the maximum number of buffered I/O's performed by the job. The return descriptor must point to an integer variable.

JAMS_K_MAX_CPUTIM

Returns the maximum amount of CPU time, in 10 millisecond units, used by the job. The return descriptor must point to an integer variable.

JAMS_K_MAX_DIOCNT

Returns the maximum number of direct I/Os performed by the job. The return descriptor must point to an integer variable.

JAMS_K_MAX_PAGEFLTS

Returns the maximum number of page faults which incurred by the job. The return descriptor must point to an integer variable.

JAMS_K_MAX_PGFLPEAK

Returns the maximum peak page file usage of the job. The return descriptor must point to an integer variable.

JAMS_K_MAX_WSPEAK

Returns the maximum peak working set of the job. The return descriptor must point to an integer variable.

JAMS_K_MIN_BIOCNT

Returns the minimum number of buffered I/O's performed by the job. The return descriptor must point to an integer variable.

JAMS_K_MIN_CPUTIM

Returns the minimum amount of CPU time, in 10 millisecond units, used by the job. The return descriptor must point to an integer variable.

JAMS_K_MIN_DIOCNT

Returns the minimum number of direct I/Os performed by the job. The return descriptor must point to an integer variable.

JAMS_K_MIN_PAGEFLTS

Returns the minimum number of page faults which incurred by the job. The return descriptor must point to an integer variable.

JAMS_K_MIN_PGFLPEAK

Returns the minimum peak page file usage of the job. The return descriptor must point to an integer variable.

JAMS_K_MIN_WSPEAK

Returns the minimum peak working set of the job. The return descriptor must point to an integer variable.

JAMS_K_NAME

Returns the name of the job. If the job was submitted via a Setup Definition, the name of the Setup is returned, otherwise, the name of the job is returned. A job name may be up to 39 characters in length. The return descriptor must point to a character string variable. If the context is from a call to *JAMS_SCHEDULE_GET_DEPEND* then this item code returns the name of the Job or Variable involved in the dependency.

JAMS_K_NODENAME

Returns the node name on which the job executed. The node name may be up to 64 characters long. The return descriptor must point to a character string variable.

JAMS_K_NONWORKDAY_OPTION

Returns the Non-workday flag of a Setup definition. This flag will be "S", "I" or "D". The return descriptor must point to a character string variable.

JAMS_K_NOTIFY_SEVERITY

Returns the Notify Severity flag of a Job definition. This flag will be "S", "I", "W", "E" or "F". The return descriptor must point to a character string variable.

JAMS_K_OPER_CLASSES

Returns the Operator classes specified in the job's System I.D. or, if the job is unregistered, in the *JAMS* configuration. The list of classes may be up to 256 characters in length. The return descriptor must point to a character string variable.

JAMS_K_OVERRIDE_NAME

Returns the override name given to a Job. This is the name supplied by the /NAME= qualifier of the SUBMIT command. The return descriptor must point to a character string variable.

JAMS_K_PAGEFLTS

Returns the number of page faults which occurred during the execution of the job. The return descriptor must point to an integer variable.

JAMS_K_PGFLPEAK

Returns the peak page file usage of the job. The return descriptor must point to an integer variable.

JAMS_K_PID

Returns the VMS process ID of the job's process. The return descriptor must point to an integer variable.

JAMS_K_PRINT_DIR

Returns the Report Location as specified in the System Definition. The return descriptor must point to a string variable.

JAMS_K_QUEUE_NAME

Returns the name of the VMS queue in which the job executed. The return descriptor must point to a string variable.

JAMS_K_RECOVERY_SEVERITY

Returns the Recovery Severity flag of a Job definition. This flag will be "S", "I", "W", "E" or "F". The return descriptor must point to a character string variable.

JAMS_K_REPLY_USERS

Returns the VMS Reply username list specified in the job's System I.D. or, if the job is unregistered, in the JAMS configuration. The list of usernames may be up to 256 characters in length. The return descriptor must point to a character string variable.

JAMS_K_RESTARTABLE

Returns the Restartable flag of a Job definition. This flag will be "Y" or "N". The return descriptor must point to a character string variable.

JAMS_K_RESTART_COUNT

Returns the number of times that this job restarted. The return descriptor must point to an integer variable.

JAMS_K_RESTART_NUMBER

Returns the number of this restart. The return descriptor must point to an integer variable.

JAMS_GET_DATA

JAMS_K_RESUBMIT_BASE

Returns the Re-submit from flag of a Setup definition. This flag will be “S”, “E” or “T”. The return descriptor must point to a character string variable.

JAMS_K_RESUBMIT_DELAY

Returns the Re-submit delay for a Setup Definition. The return descriptor must point to an integer variable.

JAMS_K_RESUBMIT_END

Returns the time of day when Re-submits should stop for a Setup Definition. The time of day is expressed as the number of seconds since midnight. A negative number represents a missing time. The return descriptor must point to an integer variable.

JAMS_K_RON

Returns the *JAMS* run occurrence number (RON) of this job. Each job is assigned a unique RON which will never repeat. The return descriptor must point to an integer variable.

JAMS_K_RUN_UNDER

Returns the VMS username under which this job ran. The return descriptor must point to a character string variable.

JAMS_K_SATISFIED

Returns an integer which will have the value 1 if the dependency was satisfied in the simulation or zero if the dependency was not satisfied. The return descriptor must point to an integer variable.

JAMS_K_SATISFIED_TIME

Returns a VMS quadword date and time which represents the date and time when the dependency was satisfied. Zero is returned if the dependency was not satisfied. The return descriptor must point to an 8 byte variable.

JAMS_K_SCHEDULE_TYPE

Returns a numeric code which represents what type of scheduled Job this is. Possible values are:

Symbol	Description
JAMS_K_PRECHECK	The Job was submitted as a pre-check for another job in the schedule.
JAMS_K_RECOVERY	The Job was submitted as a recovery job for another job in the schedule.
JAMS_K_REPEATED	The Job was submitted as the result of the completion of a Setup which is resubmitted periodically.
JAMS_K_SCHEDULED	The Job is an normal scheduled Job.
JAMS_K_TRIGGERED	The Job was submitted as the result of a Trigger firing during the simulation.

The return descriptor must point to an integer variable.

JAMS_K_SCHED_DATE

Returns the the text of a Setup Definitions Scheduled Date. The return descriptor must point to a character string variable.

JAMS_K_SCHED_EXCEPT

Returns the the text of a Setup Definitions Except for Date. The return descriptor must point to a character string variable.

JAMS_K_SCHED_TIME

Returns the scheduled time of the job as a VMS quadword date and time. The return descriptor must point to an 8 byte variable.

JAMS_K_SETUP_DESC_LN1, JAMS_K_SETUP_DESC_LN2

If this job was submitted via a Setup definition, returns line one or line two of the Setup description , otherwise returns spaces. A description may be up to 50 characters in length. The return descriptor must point to a character string variable.

JAMS_K_SETUP_NAME

If this job was submitted via a Setup definition, returns the name of the Setup, otherwise returns spaces. A Setup name may be up to 39 characters in length. The return descriptor must point to a character string variable.

JAMS_K_SINCE_DESC_LN1, JAMS_K_SINCE_DESC_LN2

Returns line one or line two of the description of the Since Job. The description may be up to 50 characters in length. The return descriptor must point to a character string variable.

JAMS_K_SINCE_NAME

Returns the name of the Since Job. The name may be up to 39 characters in length. The return descriptor must point to a character string variable.

JAMS_K_SINCE_TIME

Returns a VMS quadword date and time which represents the date and time after which the dependency had to be satisfied. For example, if a dependency says “JOBA must have completed since the last time that JOBB completed with a status of I or better”, then the date and time returned by this item code will be the last time that JOBB completed with a status of I or better. The return descriptor must point to an 8 byte variable.

JAMS_K_START_TIME

Returns the starting time of the job as a VMS quadword date and time. The return descriptor must point to an 8 byte variable.

JAMS_K_SUBMITTED_BY

Returns the VMS username which submitted this job. The return descriptor must point to a character string variable.

JAMS_K_SUBMIT_FROM

Returns the time of day which starts this Job or Setups Submit range. The time of day is expressed as the number of seconds since midnight. A negative number represents a missing time. The return descriptor must point to an integer variable.

JAMS_K_SUBMIT_HOLD

Returns the Submit on Hold flag of a Job or Setup definition. This flag will be “Y” or “N”. The return descriptor must point to a character string variable.

JAMS_K_SUBMIT_TIME

Returns the submit time of the job as a VMS quadword date and time. The return descriptor must point to an 8 byte variable.

JAMS_K_SUBMIT_TO

Returns the time of day which ends this Job or Setups Submit range. The time of day is expressed as the number of seconds since midnight. A negative number represents a missing time. The return descriptor must point to an integer variable.

JAMS_K_SYSTEM_DESC

Returns the description of this Job's System. A description can be up to 50 characters long. The return descriptor must point to a character string variable.

JAMS_K_SYSTEM_ID

Returns the System I.D. of this Job. A System I.D. can be up to 12 characters long. The return descriptor must point to a character string variable.

JAMS_K_VOLUMES

Returns number of volumes mounted by this job. The return descriptor must point to an integer variable.

JAMS_K_WSPEAK

Returns the peak working set of the job. The return descriptor must point to an integer variable.

RETURN VALUES		
	JAMS_INVCTX	Invalid context passed in.
	JAMS_INVDATAID	Invalid data identifier.
	JAMS_INVRETDSC	Invalid return descriptor.

JAMS_GET_JOB_STATUS—Get a Jobs Status Text.

The JAMS_GET_JOB_STATUS routine is used to get the value of a Job's status text.

FORMAT	JAMS_GET_JOB_STATUS	<i>reserved,</i> <i>value</i>
---------------	----------------------------	----------------------------------

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
----------------	---

ARGUMENTS	<i>reserved</i> VMS Usage: reserved type: reserved access: read only mechanism: by value This argument is reserved for future use. <i>value</i> VMS Usage: string type: string access: write only mechanism: by descriptor This argument is a descriptor which points to the buffer where the Job's current status text is returned.
------------------	---

DESCRIPTION	The <i>JAMS_GET_JOB_STATUS</i> routine can be called when you want to get a Job's current status text. The Job whose status is returned is the one which is running the program which calls this routine.
--------------------	---

JAMS_HISTORY_FREE

The JAMS_HISTORY_FREE routine is used to free memory which has been allocated by a call to JAMS_HISTORY_INIT.

FORMAT	JAMS_HISTORY_FREE <i>context</i>
---------------	---

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
----------------	---

ARGUMENTS	<i>context</i> VMS Usage: context type: pointer access: read/write mechanism: by reference This argument is the history context which was returned from a call to JAMS_HISTORY_INIT.
------------------	--

DESCRIPTION	This routine is used to free memory allocated by the JAMS_HISTORY_INIT routine. This routine destroys the history context.
--------------------	--

JAMS_HISTORY_GET

The JAMS_HISTORY_GET routine is used to set a history context to reference the first, or next, history record which matches the selection criteria.

FORMAT	JAMS_HISTORY_GET	<i>context</i>
---------------	-------------------------	----------------

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
----------------	---

ARGUMENTS	<i>context</i> VMS Usage: context type: pointer access: read only mechanism: by value This argument is the history context which was returned from a call to JAMS_HISTORY_INIT.
------------------	---

DESCRIPTION	This routine is used to set a history context to reference the first, or next, history record which matches the selection criteria. This history context must be initialized by calling JAMS_HISTORY_INIT. The history context can then be passed to JAMS_GET_DATA to obtain information from the context.
--------------------	--

RETURN VALUES	JAMS_ENDCTX	End of context stream.
--------------------------	-------------	------------------------

JAMS_HISTORY_INIT

The JAMS_HISTORY_INIT routine is used to search the JAMS history database.

FORMAT **JAMS_HISTORY_INIT** *context, item-list*

RETURNS VMS Usage: **cond_value**
 type: **longword (unsigned)**
 access: **write only**
 mechanism: **by value**

ARGUMENTS ***context***
 VMS Usage: **context**
 type: **pointer**
 access: **write only**
 mechanism: **by reference**
 This argument is a pointer which will be used to return the history context which is created and initialized by this routine.

item-list

VMS Usage: **item_list_3**
 type: **longword (unsigned)**
 access: **read only**
 mechanism: **by reference**

This argument is a pointer to an item list. An item list is an array of item descriptors, each of which specifies one input or output data item.

The item list is terminated by an item descriptor with a value of zero in the item code and buffer length fields.

The following item codes are supported by the JAMS_HISTORY_INIT routine:

JAMS_K_END_DATE

The JAMS_K_END_DATE item code is used to pass the date and time of the end of the time range from which history will be returned. The date and time must be passed as a standard VMS quadword date.

JAMS_K_JOB_NAME

The JAMS_K_JOB_NAME item code is used to pass a Job Name selection specification. Wildcard characters are allowed. Only history of Jobs which match this selection specification will be included. Omitted this item is the same as specifying an asterisk.

JAMS_K_OPTIONS

The *JAMS_K_OPTIONS* item code is used to pass a longword which specifies the options which should be activated during this call. Each bit in the longword corresponds to one option. You can initialize this longword by ORing (or adding) together one or more of the following symbols.

Symbol	Function
JAMS_M_SUBMITTED	Include jobs which were submitted in the specified time range.
JAMS_M_COMPLETED	Include jobs which completed in the specified time range.
JAMS_M_STARTED	Include jobs which started in the specified time range.

JAMS_K_SETUP_NAME

The *JAMS_K_SETUP_NAME* item code is used to pass a Setup Name selection specification. Wildcard characters are allowed. Only history of Setups which match this selection specification will be included. A history record of a Job which was not submitted via a Setup will only match an asterisk. Omitted this item is the same as specifying an asterisk.

JAMS_K_SEVERITY

The *JAMS_K_SEVERITY* item code is used to pass a longword which specifies the final status severities which should be returned. Only history records from job which termination with a severity which matches one of the specified severities will be returned. You can initialize this longword by ORing (or adding) together one or more of the following symbols.

JAMS_M_SUCCESS
JAMS_M_INFO
JAMS_M_WARNING
JAMS_M_ERROR
JAMS_M_FATAL

JAMS_K_START_DATE

The *JAMS_K_START_DATE* item code is used to pass the date and time of the start of the time range from which history will be returned. The date and time must be passed as a standard VMS quadword date.

JAMS_K_SYSTEM_ID

The *JAMS_K_SYSTEM_ID* item code is used to pass a System I.D. selection specification. Wildcard characters are allowed. Only history from Jobs and Setups in a System which matches this selection specification will be included.

DESCRIPTION

The *JAMS_HISTORY_INIT* routine is the first step in searching the *JAMS* job execution history information. After successfully calling *JAMS_HISTORY_INIT*, you call *JAMS_HISTORY_GET* to return a History Context which can then be passed to *JAMS_GET_DATA* to get data from the context.

JAMS3000.COB is an example of using the JAMS_HISTORY_INIT routine. JAMS3000.COB can be found in JAMS_EXAMPLES.

RETURN VALUES

JAMS_INVIL	Invalid item list.
SS\$_NOPRIV	Insufficient privileges

JAMS_GET_VARIABLE—Return a Variable's current value

The JAMS_GET_VARIABLE routine is used to return the current value of a JAMS Variable.

FORMAT	JAMS_GET_VARIABLE	<i>variable-name,</i> <i>variable-value</i>
---------------	--------------------------	--

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
----------------	---

ARGUMENTS	<i>variable-name</i> VMS Usage: string type: string access: read only mechanism: by descriptor The name of the Variable whose value you want returned. You can access a Variable on a remote node by specifying the remote node name and access control information in the standard DECnet format.
	<i>variable-value</i> VMS Usage: various type: various access: write only mechanism: by descriptor A descriptor of the data item into which the Variables value will be returned.

DESCRIPTION	The <i>JAMS_GET_VARIABLE</i> routine can be called when you want to obtain the current value of a JAMS Variable.
--------------------	--

RETURN VALUES	JAMS_VARNOTFOU Variable not found. JAMS_INVRETDSC Invalid return descriptor.
----------------------	---

JAMS_SCHCALLUSER—User-Written procedure

The *JAMS_SCHCALLUSER* routine is a user-written procedure which is called by the *JAMS_SCHEDULE* process when events occur.

FORMAT **JAMS_SCHCALLUSER** *type, context*

RETURNS VMS Usage: **cond_value**
 type: **longword (unsigned)**
 access: **write only**
 mechanism: **by value**

ARGUMENTS

type

VMS Usage: **type**
 type: **longword (unsigned)**
 access: **read only**
 mechanism: **by reference**

Identifies the reason for calling your routine. The *type* parameter will contain one of the following codes:

JAMS_K_SCHSTART	JAMS_SCHEDULE process is starting up.
JAMS_K_SCHSTOP	JAMS_SCHEDULE process is stopping.
JAMS_K_JOBSTART	A Job is starting to execute.
JAMS_K_JOBEND	A Job has completed.

While these are the only possible codes for this release of *JAMS*, you should code your procedure to ignore unrecognized codes because future versions of *JAMS* may call your procedure for other reasons, with new codes.

context

VMS Usage: **context**
 type: **longword (unsigned)**
 access: **read only**
 mechanism: **by reference**

This is a context variable which you must use when calling the *JAMS_GET_DATA* routine.

DESCRIPTION

The *JAMS_SCHCALLUSER* routine is a user-written procedure. This procedure allows you to gain control when certain events occur. This procedure will be called when the *JAMS_SCHEDULE* process starts or stops and when a job starts executing and completes.

Your routine can call the *JAMS_GET_DATA* routine to obtain detailed information concerning the event which triggered the call. This routine must complete relatively quickly because *JAMS_SCHEDULE* process is stalled until this routine completes.

JAMS_SCHCALLUSER

This routine may be written in any language which adheres to the OpenVMS Common Language Environment. You must link your routine into a sharable image. The sharable image must be named JAMS_SCHCALLUSER.EXE and be located in SYS\$SHARE or you must define the logical name JAMS_SCHCALLUSER to point to your sharable image. The JAMS_SCHCALLUSER logical name must be defined /EXECUTIVE and /SYSTEM.

The following commands illustrate how to compile, link and install a *JAMS_SCHCALLUSER* routine on a VAX.

```
$ COBOL JAMS_SCHCALLUSER.COB
$ LINK/EXE=JAMS_SCHCALLUSER.EXE SYS$INPUT/OPTION/SHARE
JAMS_SCHCALLUSER.OBJ
UNIVERSAL=JAMS_SCHCALLUSER
JAMS_EXE:JAMSSHR.EXE/SHARE
Ctrl/Z
$ COPY JAMS_SCHCALLUSER.EXE SYS$COMMON:[SYSLIB]
$ MCR JAMS_EXE:JAMS_MASTER.EXE STOP SCHEDULE
$ MCR JAMS_EXE:JAMS_MASTER.EXE START SCHEDULE
```

The following commands illustrate how to compile, link and install a *JAMS_SCHCALLUSER* routine on an Alpha.

```
$ COBOL JAMS_SCHCALLUSER.COB
$ LINK/EXE=JAMS_SCHCALLUSER.EXE SYS$INPUT/OPTION/SHARE
JAMS_SCHCALLUSER.OBJ
SYMBOL_VECTOR=(JAMS_SCHCALLUSER=PROCEDURE)
JAMS_EXE:JAMSSHR.EXE/SHARE
Ctrl/Z
$ COPY JAMS_SCHCALLUSER.EXE SYS$COMMON:[SYSLIB]
$ MCR JAMS_EXE:JAMS_MASTER.EXE STOP SCHEDULE
$ MCR JAMS_EXE:JAMS_MASTER.EXE START SCHEDULE
```

There are examples of *JAMS_SCHCALLUSER* routines in JAMS_EXAMPLES:.

JAMS_SCHEDULE_FREE

The JAMS_SCHEDULE_FREE routine is used to free memory which has been allocated by a call to JAMS_SCHEDULE_INIT.

FORMAT **JAMS_SCHEDULE_FREE** *context*

RETURNS VMS Usage: **cond_value**
 type: **longword (unsigned)**
 access: **write only**
 mechanism: **by value**

ARGUMENTS ***context***
 VMS Usage: **context**
 type: **pointer**
 access: **read/write**
 mechanism: **by reference**
 This argument is the schedule context which was returned from a call to JAMS_SCHEDULE_INIT.

DESCRIPTION This routine is used to free memory allocated by the JAMS_SCHEDULE_INIT routine. This routine destroys the schedule context.

JAMS_SCHEDULE_GET_DEPEND

The JAMS_SCHEDULE_GET_DEPEND routine is used to return the first or next dependency context from a job context.

FORMAT	JAMS_SCHEDULE_GET_DEPEND	<i>job-context,</i> <i>depend-context</i>
--------	--------------------------	--

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
---------	---

ARGUMENTS	<i>job-context</i> VMS Usage: context type: pointer access: read only mechanism: by value This argument is the job context which was returned from a call to JAMS_SCHEDULE_GET_JOB. <i>depend-context</i> VMS Usage: context type: pointer access: write only mechanism: by reference This argument is a pointer which will be used to return the dependency context.
-----------	--

DESCRIPTION	This routine is used to return the first, or next, dependency context from a job context. The dependency context can then be passed to JAMS_GET_DATA to obtain information from the context.
-------------	--

RETURN VALUES	JAMS_ENDCTX	End of context stream.
------------------	-------------	------------------------

JAMS_SCHEDULE_GET_JOB

The JAMS_SCHEDULE_GET_JOB routine is used to return the first or next job context from a schedule context.

FORMAT	JAMS_SCHEDULE_GET_JOB	<i>context, job-context</i>
---------------	------------------------------	-----------------------------

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
----------------	---

ARGUMENTS	<i>context</i> VMS Usage: context type: pointer access: read only mechanism: by value This argument is the schedule context which was returned from a call to JAMS_SCHEDULE_INIT. <i>job-context</i> VMS Usage: context type: pointer access: write only mechanism: by reference This argument is a pointer which will be used to return the job context.
------------------	---

DESCRIPTION	This routine is used to return the first, or next, job context from a schedule context. The job context can then be passed to JAMS_GET_DATA to obtain information from the context or, it can be passed to JAMS_SCHEDULE_GET_DEPEND to get a dependency context.
--------------------	--

RETURN VALUES	JAMS_ENDCTX	End of context stream.
--------------------------	-------------	------------------------

JAMS_SCHEDULE_INIT

The JAMS_SCHEDULE_INIT routine is used to create and initialize a simulated job schedule.

FORMAT	JAMS_SCHEDULE_INIT <i>context, item-list</i>
---------------	---

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
----------------	---

ARGUMENTS	<i>context</i> VMS Usage: context type: pointer access: write only mechanism: by reference This argument is a pointer which will be used to return the schedule context which is created and initialized by this routine. <i>item-list</i> VMS Usage: item_list_3 type: longword (unsigned) access: read only mechanism: by reference This argument is a pointer to an item list. An item list is an array of item descriptors, each of which specifies one input or output data item. The item list is terminated by an item descriptor with a value of zero in the item code and buffer length fields. The following item codes are supported by the <i>JAMS_SCHEDULE_INIT</i> routine: JAMS_K_END_DATE The <i>JAMS_K_END_DATE</i> item code is used to pass the date and time of the end of the simulated schedule. The date and time must be passed as a standard VMS quadword date. JAMS_K_JOB_NAME The <i>JAMS_K_JOB_NAME</i> item code is used to pass a Job Name selection specification. Wildcard characters are allowed. When the initial schedule is created, only Jobs which match this selection specification will be included. If this item is omitted, all Jobs are included.
------------------	---

JAMS_K_OPTIONS

The *JAMS_K_OPTIONS* item code is used to pass a longword which specifies the options which should be activated during this call. Each bit in the longword corresponds to one option. You can initialize this longword by ORing (or adding) together one or more of the following symbols.

Symbol	Function
JAMS_M_CURRENT	Use the Jobs which are currently scheduled as the basis for the simulation. If this option is used, you should not include start and end dates in the item list.
JAMS_M_INCLUDE_PLANNED	If this option is used, Setups which are <i>not</i> automatically submitted will be included in the simulated schedule (provided that they match other selection criteria).
JAMS_M_INCLUDE_REPEATING	If this option is used, Setups which are automatically resubmitted will have all occurrences shown in the simulated schedule.
JAMS_M_INCLUDE_TRIGGERED	If this option is used, the simulation will include jobs submitted by the simulated firing of Trigger definitions.

JAMS_K_SETUP_NAME

The *JAMS_K_SETUP_NAME* item code is used to pass a Setup Name selection specification. Wildcard characters are allowed. When the initial schedule is created, only Setups which match this selection specification will be included. If this item is omitted, all Setups are included.

JAMS_K_START_DATE

The *JAMS_K_START_DATE* item code is used to pass the date and time of the start of the simulated schedule. The date and time must be passed as a standard VMS quadword date.

JAMS_K_SYSTEM_ID

The *JAMS_K_SYSTEM_ID* item code is used to pass a System I.D. selection specification. Wildcard characters are allowed. When the initial schedule is created, only Jobs and Setups in a System which matches this selection specification will be included.

DESCRIPTION

The JAMS_SCHEDULE_INIT routine is the first step in creating a simulated schedule. This simulated schedule can be used for reporting or capacity planning purposes. After successfully calling JAMS_SCHEDULE_INIT, you can call JAMS_SCHEDULE_SORT to sort the jobs in the simulated schedule. You call JAMS_SCHEDULE_GET_JOB to return a Job Context which can then be passed to JAMS_GET_DATA to get data from the context.

JAMS2100.COB is an example of using the JAMS_SCHEDULE_INIT routine. JAMS2100.COB can be found in SYS\$EXAMPLES.

**RETURN
VALUES**

JAMS_INVIL	Invalid item list.
SS\$_NOPRIV	Insufficient privileges

JAMS_SCHEDULE_SORT

The JAMS_SCHEDULE_SORT routine is used to sort the Jobs in a schedule context.

FORMAT **JAMS_SCHEDULE_SORT** *context, sort-code*

RETURNS VMS Usage: **cond_value**
 type: **longword (unsigned)**
 access: **write only**
 mechanism: **by value**

ARGUMENTS ***context***
 VMS Usage: **context**
 type: **pointer**
 access: **read only**
 mechanism: **by value**
 This argument is the schedule context which was returned from a call to JAMS_SCHEDULE_INIT.

sort-code
 VMS Usage: **code**
 type: **longword**
 access: **read only**
 mechanism: **by value**
 This argument is a sort code which specifies how the Jobs should be sorted. Valid values are:

Symbol	Sort Order
JAMS_K_SORT_SYSTEM_ID	Sorts by System I.D., Setup Name, Job Name and Scheduled Date/Time.
JAMS_K_SORT_JOB_NAME	Sorts by Job Name, Setup Name and Scheduled Date/Time.
JAMS_K_SORT_SCHEDULED_DATE	Sorts by Scheduled Date/Time, Setup Name, and Job Name.
JAMS_K_SORT_ESTIMATED_DATE	Sorts by Estimated Start time, Setup Name and Job Name.

DESCRIPTION The JAMS_SCHEDULE_SORT routine is used to sort the Jobs in a simulated schedule.

JAMS_SET_DATA—Return data to the JAMS_SCHEDULE process.

The JAMS_SET_DATA routine is used to return data to the JAMS_SCHEDULE process.

FORMAT	JAMS_SET_DATA	<i>context,</i> <i>data-ident1, return-desc1,</i> <i>[data-ident2, data-desc2]...</i>
---------------	----------------------	---

RETURNS

ARGUMENTS *context*
VMS Usage: **context**
type: **longword (unsigned)**
access: **read only**
mechanism: **by reference**
This is the context variable which was passed to your routine by *JAMS*.
The *JAMS_SET_DATA* routine can be called only when a context has been
established and passed to your code via a *JAMS* calluser routine.

data-identN

```
VMS Usage:  longword
type:       longword
access:     read only
mechanism:  by value
```

The *data-ident* parameter identifies a data element which you want to return to the JAMS_SCHEDULE process.

data-descN

VMS Usage: **various**
type: **various**
access: **write only**
mechanism: **by descriptor**

The *data-desc* parameter is a descriptor which points to the location of the variable which is to be returned to the JAMS_SCHEDULE process.

DESCRIPTION	The <i>JAMS_SET_DATA</i> routine can be called from your <i>JAMS_SCHCALLUSER</i> routine to modify data associated with a job completion. When calling <i>JAMS_SET_DATA</i> you pass one or more pairs of <i>data-ident</i> and <i>data-desc</i> parameters.
--------------------	---

The data identifiers supported vary based on the type of calluser context. *JAMS* can call your code with the following context types:

JAMS_K_SCHSTART	JAMS_SCHEDULE process is starting up.
JAMS_K_SCHSTOP	JAMS_SCHEDULE process is stopping.
JAMS_K_JOBSTART	A Job is starting to execute.
JAMS_K_JOBEND	A Job has completed.

The following data identifiers are supported by the *JAMS_SET_DATA* routine when the context type is *JAMS_K_JOBEND*:

JAMS_K_FINAL_STATUS

Allows you to change the final status of a job. You could, for example, scan a job's .LOG file for a specific string and change a good status to a bad status if the string was found. The data descriptor must point to an integer variable.

JAMS_K_MAIL_ADR

Allows you to change the list of VMS mail addresses which will be notified if this Job ended with a bad status. The list of addresses may be up to 256 characters in length. The data descriptor must point to a character string variable.

JAMS_K_OPER_CLASSES

Allows you to change the list of VMS OPCOM operator classes which will receive an OPCOM message if this Job ended with a bad status. The list of classes may be up to 256 characters in length. The data descriptor must point to a character string variable.

JAMS_K_REPLY_USERS

Allows you to change the list of VMS username which will receive a broadcast message if this Job ended with a bad status. The list of usernames may be up to 256 characters in length. The data descriptor must point to a character string variable.

RETURN VALUES

JAMS_INVCTX	Invalid context passed in.
JAMS_INVDATAID	Invalid data identifier.
JAMS_INVRETDSC	Invalid data descriptor.

JAMS_SET_JOB_STATUS—Set a Job's Status Text.

The JAMS_SET_JOB_STATUS routine is used to set the value of a Job's status text.

FORMAT	JAMS_SET_JOB_STATUS <i>reserved,</i> <i>value,</i> <i>[monitor]</i>
---------------	--

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
----------------	---

ARGUMENTS	<i>reserved</i>
------------------	------------------------

	VMS Usage: reserved type: reserved access: read only mechanism: by value
--	---

This argument is reserved for future use.

value

	VMS Usage: string type: string access: read only mechanism: by descriptor
--	--

This is the new value for the Job's status text.

monitor

	VMS Usage: boolean type: longword access: read only mechanism: by reference
--	--

This argument is used to specify whether or not the status text should be displayed in the *JAMS* Job Monitor. If omitted, the monitor status is left unchanged. This argument must be a longword passed by reference. If the value of the longword is zero, the status will not be monitored. If the value is 1, the status text is monitored.

DESCRIPTION

The *JAMS_SET_JOB_STATUS* routine can be called when you want to set a Job's status text. The Job whose status is changed is the one which is running the program which calls this routine.

JAMS_SET_VARIABLE—Updates a Variable's current value

The JAMS_SET_VARIABLE routine is used to update the current value of a JAMS Variable.

FORMAT	JAMS_SET_VARIABLE	<i>variable-name</i> , <i>[variable-value]</i> <i>[options]</i>
---------------	--------------------------	---

RETURNS	VMS Usage: cond_value type: longword (unsigned) access: write only mechanism: by value
----------------	---

ARGUMENTS	<i>variable-name</i> VMS Usage: string type: string access: read only mechanism: by descriptor
------------------	---

The name of the Variable whose value you want to modify. You can access a Variable on a remote node by specifying the remote node name and access control information in the standard DECnet format.

variable-value

VMS Usage: **various**
type: **various**
access: **read only**
mechanism: **by descriptor**

A descriptor of the data item which will be the new value of the Variable. This may also be a value which is added to or subtracted from the Variable's current value, depending on the options specified.

options

VMS Usage: **bit mask**
type: **longword (unsigned)**
access: **read only**
mechanism: **by reference**

A Longword options bit mask. Each bit in the longword corresponds to one option. You can initialize this longword by ORing (or adding) together one or more of the following symbols.

JAMS_SET_VARIABLE

Symbol	Function
JAMS_M_INCREMENT	The Variable's value should be incremented by the value passed in the variable-value argument.
JAMS_M_DECREMENT	The Variable's value should be decremented by the value passed in the variable-value argument.
JAMS_M_CURRENT	Valid only for date and/or time Variables. The Variable should be set to the current date and/or time. The variable-value argument is ignored.

DESCRIPTION

The *JAMS_SET_VARIABLE* routine can be called when you want to modify the current value of a *JAMS* Variable.

RETURN VALUES

JAMS_VARNOTFOU	Variable not found.
JAMS_INVRETDSC	Invalid return descriptor.

JAMS_SUBMIT_MENU—Display a Menu

The JAMS_SUBMIT_MENU routine provides an entry into JAMS at the menu level.

FORMAT **JAMS_SUBMIT_MENU** *item-list*

RETURNS VMS Usage: **cond_value**
 type: **longword (unsigned)**
 access: **write only**
 mechanism: **by value**

ARGUMENTS ***item-list***

VMS Usage: **item_list_3**
 type: **longword (unsigned)**
 access: **read only**
 mechanism: **by reference**

This argument is a pointer to an item list. An item list is an array of item descriptors, each of which specifies one input or output data item.

The item list is terminated by an item descriptor with a value of zero in the item code and buffer length fields.

The following item codes are supported by the *JAMS_SUBMIT_MENU* routine:

JAMS_K_MENUNAME

The *JAMS_K_MENUNAME* item code is used to pass the name of a Menu Definition which should be used as the initial menu.

You cannot specify both *JAMS_K_MENUNAME* and *JAMS_K_SYSTEMID* in the same item list.

JAMS_K_REMOTE_NODE

The *JAMS_K_REMOTE_NODE* item code is used to pass then name and access control information of a remote node where the menus are defined. The jobs selected will be submitted on the remote node. The format of the specification is the standard DECnet format, nodename"username password". If you have proxy access to the remote node, you only need to specify the nodename.

JAMS_K_SYSTEMID

The *JAMS_K_SYSTEMID* item code is used to pass the name of a System Definition. A menu is constructed which consists of all of the Job and Job Setup definitions for the indicated System.

You cannot specify both *JAMS_K_MENUNAME* and *JAMS_K_SYSTEMID* in the same item list.

JAMS_SUBMIT_MENU

JAMS_K_OPTIONS

The *JAMS_K_OPTIONS* item code is used to pass a longword which specifies the options which should be activated during this call. Each bit in the longword corresponds to one option. You can initialize this longword by ORing (or adding) together one or more of the following symbols.

Symbol	Function
JAMS_M_DEBUG	The Job should be submitted in debug mode.
JAMS_M_NOPROMPT	The user should not be prompted for parameter values or report overrides.
JAMS_M_NOSUBMIT	The Job should not be submitted, instead a parsed job file is created in the current default directory with the extension .JAMS.
JAMS_M_USE_SYMBOLS	DCL symbols should be used to determine the default value of parameters.
JAMS_M_SET_SYMBOLS	A DCL symbol for each of the Job's parameters should be defined after the Job is submitted.
JAMS_M_ENABLE_CONNECTIONS	Adds the Gold/C function key to the submit menu to allow the user to establish remote connections.

DESCRIPTION

JAMS_SUBMIT_MENU can be called when you want to provide a menu of batch jobs which may be submitted. This routine requires one argument, an item list.

When *JAMS_SUBMIT_MENU* is called, *JAMS* will take control of the terminal and display the selected menu of batch jobs and/or menus. The user is able to select and submit jobs. Control is not returned until the user exits the initial menu.

RETURN VALUES

JAMS_CANTSUBMIT	Errors encountered while trying to submit job.
JAMS_INVIL	Invalid item list.
SS\$_NOPRIV	Insufficient privileges

JAMS_SUBMIT_JOB—Submit a Batch Job

The JAMS_SUBMIT_JOB routine is used to submit a single batch job via the JAMS batch job submission sub-system.

FORMAT **JAMS_SUBMIT_JOB** *item-list*

RETURNS VMS Usage: **cond_value**
 type: **longword (unsigned)**
 access: **write only**
 mechanism: **by value**

ARGUMENTS ***item-list***

VMS Usage: **item_list_3**
 type: **longword (unsigned)**
 access: **read only**
 mechanism: **by reference**

This argument is a pointer to an item list. An item list is an array of item descriptors, each of which specifies one input or output data item.

The item list is terminated by an item descriptor with a value of zero in the item code and buffer length fields.

The following item codes are supported by the *JAMS_SUBMIT_JOB* routine:

JAMS_K_DEPEND_ENTRY

The *JAMS_K_DEPEND_ENTRY* item code is used to pass the VMS queue entry number of another job which the job being submitted should depend on. The job being submitted will remain in a pending state until the job specified with this item code has completed successfully.

You can specify this item code up to 32 times in the same item list so a job which you submit can depend on up to 32 other jobs.

JAMS_K_JOBNAME

The *JAMS_K_JOBNAME* item code is used to pass the name of the Job which is to be submitted. The *JAMS_K_JOBNAME* item code may be used only once in an item list.

In most cases, the order of the item codes in an item list is insignificant. The order is significant with the *JAMS_K_JOBNAME* and *JAMS_K_SETUPNAME* item codes. If you pass only one of these two item codes, the position in the item list is insignificant. However, if you pass both of these item codes the position is significant, and both items must specify the same value.

What you are really saying when you pass both of these item codes is “This name is either a Job name or a Setup name.” If the *JAMS_K_JOBNAME* item code is first in the item list, then a Job with the specified name is searched for. If the Job is not found, then a Setup is searched for. Specify the *JAMS_K_SETUPNAME* item code first if you want to search for a Setup before searching for a Job.

JAMS_K_OPTIONS

The *JAMS_K_OPTIONS* item code is used to pass a longword which specifies the options which should be activated during this call. Each bit in the longword corresponds to one option. You can initialize this longword by ORing (or adding) together one or more of the following symbols.

Symbol	Function
JAMS_M_DEBUG	The Job should be submitted in debug mode.
JAMS_M_NOPROMPT	The user should not be prompted for parameter values or report overrides.
JAMS_M_NOSUBMIT	The Job should not be submitted, instead a parsed job file is created in the current default directory with the extension .JAMS.
JAMS_M_USE_SYMBOLS	DCL symbols should be used to determine the default value of parameters.
JAMS_M_SET_SYMBOLS	A DCL symbol for each of the Job's parameters should be defined after the Job is submitted.

JAMS_K_PARAM_NAME

The *JAMS_K_PARAM_NAME* item code is used to pass the name of a parameter. This item code is used form *JAMS* parameters which have a name other than P1 through P8. You can use this item code to pass the name of a parameter which is defined in the Job definition or to create a new parameter which is not defined in the Job's definition.

The item list entry must point to a valid parameter name which may, or may not be, defined in the Job definition.

The *JAMS_K_PARAM_NAME* item code must be directly followed by one of the following item codes:

- JAMS_K_PARAM_TEXT
- JAMS_K_PARAM_DATETIME
- JAMS_K_PARAM_INTEGER

These item codes are used to pass the *value* of the parameter named by the *JAMS_K_PARAM_NAME* item code.

JAMS_K_PARAM_TEXT

The *JAMS_K_PARAM_TEXT* item code is used to pass a text value to the parameter named in the preceeding *JAMS_K_PARAM_NAME* item list entry.

JAMS_K_PARAM_DATETIME

The *JAMS_K_PARAM_DATETIME* item code is used to pass a VMS quadword date & time value to the parameter named in the preceeding *JAMS_K_PARAM_NAME* item list entry.

JAMS_K_PARAM_INTEGER

The *JAMS_K_PARAM_INTEGER* item code is used to pass an integer value to the parameter named in the preceeding *JAMS_K_PARAM_NAME* item list entry.

JAMS_K_REMOTE_NODE

The *JAMS_K_REMOTE_NODE* item code is used to pass then name and access control information of a remote node where the Job to be submitted is defined. The job will be submitted on the remote node. The format of the specification is the standard DECnet format, *nodename"username password"*. If you have proxy access to the remote node, you only need to specify the *nodename*.

JAMS_K_SETUPNAME

The *JAMS_K_SETUPNAME* item code is used to pass the name of the Setup definition which is to be submitted. The *JAMS_K_SETUPNAME* item code may be used only once in an item list.

In most cases, the order of the item codes in an item list is insignificant. The order *is* significant with the *JAMS_K_JOBNAME* and *JAMS_K_SETUPNAME* item codes. If you pass only one of these two item codes, the position in the item list is insignificant. However, if you pass both of these item codes the position is significant, and both items must specify the same value.

What you are really saying when you pass both of these item codes is "This name is either a Job name or a Setup name." If the *JAMS_K_JOBNAME* item code is first in the item list, then a Job with the specified name is searched for. If the Job is not found, then a Setup is searched for. Specify the *JAMS_K_SETUPNAME* item code first if you want to search for a Setup before searching for a Job.

SJC\$ Item Codes

The *JAMS_SUBMIT_JOB* routine also supports all of the item codes supported by the *SCJ\$_ENTER_FILE* function of the *SYSSNDJBC* system service. Please refer to the *VMS System Services Reference Manual* for more information on these item codes.

DESCRIPTION

JAMS_SUBMIT_JOB can be called when you want to submit a single batch job and you want to take advantage of the advanced features of the *JAMS* batch job submission subsystem.

Using the JAMS Callable Interface

JAMS_SUBMIT_JOB

RETURN VALUES

JAMS_CANTSUBMIT	Errors encountered while trying to submit job.
JAMS_INVIL	Invalid item list.
SS\$_NOPRIV	Insufficient privileges

5

Templates and Job Parsing

This chapter explains the use of Execution Methods, Job Parsing and Template Libraries.

5.1 Command Files

Batch processing generally requires the use of command files (sometimes referred to as batch files). Most sites are already using a large number of command files for their batch processing. You can continue to use your existing batch processing procedures after installing *JAMS*. In order to take full advantage of *JAMS*, you must use the *JAMS* command file template expansion capabilities.

Using command file templates provides the following advantages over normal command files:

- Job command files are simplified and standardized.
- You can have up to 255 parameters per Job.
- Report printing information can be maintained by non-technical personnel.
- Report printing information can be changed at submit time.
- Parameter substitution may take place in data areas of command files.

5.1.1 Command File Template Expansion

In order to access the full capabilities of *JAMS* you must use command file templates. Fortunately, command file templates make batch command files very simple.

When a Job which uses templates is submitted, *JAMS* combines text modules in a text library with the Job's command file to produce a temporary, expanded command file. The template text library is defined in the Job's System definition.

The text modules in a Template library can be created, extracted and replaced with the OpenVMS LIBRARY/TEXT command.

The following is an example of a traditional batch command file:

Templates and Job Parsing

Command Files

```
$!  
$!  Print Inventory Audit Trail  
$!  
$! P1 = Starting date in the form MM/DD/YY  
$! P2 = Ending date in the form MM/DD/YY  
$! P3 = Inventory transaction Code  
$!  
$  on error then goto ERROR_ROUTINE  
$!  
$  if (P1 .eqs. "") then goto PARM_ERROR  
$  if (P2 .eqs. "") then goto PARM_ERROR  
$  if (P3 .eqs. "") then goto PARM_ERROR  
$!  
$  define PINV010_PRINT PRINT_DIR:PINV010.LIS  
$!  
$  RUN INV_EXE:PINV010.EXE  
$!  
$  print/nofeed/que=FAST_LP/FORM=2_PART/DELETE PINV010_PRINT  
$!  
$  EXIT  
$!  
$PARM_ERROR:  
$  REQUEST "Job PINV010 is aborting because of invalid parameters."  
$!  
$ERROR_ROUTINE:  
$  REQUEST "Error running PINV010, please check the log."  
$  EXIT
```

The command file needed for this Job when using command file template expansion is shown in the following example:

```
$!  
$!  Print Inventory Audit Trail  
$!  
$! P1 = Starting date in the form MM/DD/YY  
$! P2 = Ending date in the form MM/DD/YY  
$! P3 = Inventory transaction Code  
$!  
$  RUN INV_EXE:PINV010.EXE  
$!
```

You can see how much more simple the command file is which uses command file template expansion. This is because the standard portions of the command file have been removed and placed in a *JAMS* template. You can establish a standard template style for each System I.D. When a Job is submitted, *JAMS* will parse the template definition and merge the templates with the Job's command file to produce a temporary command file which is executed.

The next example shows this command file after expansion. This is the command file which is actually submitted for execution.

```
$!  
$!   JAMS Main Job Structure  
$!  
$     JAMS_RON := 341                1  
$     JAMS_JOB_NAME := PINV010  
$     JAMS_SYSTEM_ID := INVENTORY  
$     JAMS_SUBMITTED_BY := FRANK  
$     JAMS_SUBMIT_TIME := 10-JAN-1991 09:03  
$!  
$     P1 := "01/10/91"              2  
$     P2 := "01/15/91"  
$     P3 := "OPN"  
$!  
$ DEFINE PINV010_PRINT PRINT_DIR:PINV010.00000155 3  
$!  
$! 4  
$!  Print Inventory Audit Trail  
$!  
$! P1 = Starting date in the form MM/DD/YY  
$! P2 = Ending date in the form MM/DD/YY  
$! P3 = Inventory transaction Code  
$!  
$   RUN INV_EXE:PINV010.EXE  
$! 5  
$ PRINT/NOFEED/QUEUE=FAST_LP/DELETE -  
    /FORM=2_PART/COPIES=1 -  
    PINV010_PRINT  
$!
```

- 1 These standard DCL Symbols are defined for all Jobs. You can modify which Symbols are defined by modifying the JAMS_MAIN template module.
- 2 Define Symbols for all of the Job's parameters.
- 3 Define a logical name for all of the Job's report files. You can modify the definition of this logical by modifying the DEFINE_REPORTS template module.
- 4 The Job's command file is inserted. Any parameter references are replaced with the value of the parameter.
- 5 The report file is printed. You can modify the way reports are printed by modifying the PRINT template module.

5.2 How Templates are Expanded

JAMS will parse a Job if the Job's Execution Method has a *job_module* specified.

When a parsed Job is submitted, JAMS will look in the Job's System definition for the name and location of the template text library. The template text library must contain the *job_module* named in the Method definition. This module is the starting point for command file expansion. JAMS will parse this module to produce the temporary command file. Example 5-1 shows the JAMS_MAIN module from the default template library as it is after you install JAMS.

Templates and Job Parsing

How Templates are Expanded

As *JAMS* parses the text module, it replaces parameter references with their values and expands function references. Depending upon how you reference the Job's source, you may also be able to include Parameter and Function references in your Job file.

Example 5–1 JAMS_MAIN Template Module

```
$!  
$!   JAMS Main Job Structure  
$!  
$      JAMS_RON :=<<jams_ron>>1  
$      JAMS_JOB_NAME := <<jams_job_name>>  
$      JAMS_SETUP_NAME := <<jams_setup_name>>  
$      JAMS_SYSTEM_ID := <<jams_system_id>>  
$      JAMS_SUBMITTED_BY := <<jams_submitted_by>>  
$      JAMS_SUBMIT_TIME := <<jams_submit_time>>  
$!  
<%SYMBOLS (PARAMETER) %>2  
$!  
<%DEFINE_REPORTS (LOGICAL) %>3  
$!  
<%JAMS_JOB%>4  
$!  
<%SET_EXPIRE%>5  
$!  
<%PRINT_ALL (PRINT_ALL) %>6  
$!
```

- 1** Define DCL Symbols for some of the *JAMS* predefined parameters.
 - 2** Expand the text module SYMBOLS once for each of the Job's parameters.
 - 3** Expand the text module DEFINE_REPORTS once for each of the Job's Reports.
 - 4** Insert the Job's command file here.
 - 5** Expand the text module SET_EXPIRE once for each of the Job's reports.
 - 6** Expand the text module PRINT_ALL once for each report which has not been printed.
-

5.2.1 Simple Parameter References

A Simple Parameter Reference has the following format:

```
<<parameter-name[(format)]>>
```

The parameter name must be either a parameter defined in the Job Definition, the name of a *JAMS* Variable Definition or a *JAMS* predefined parameter. *JAMS* predefined parameters contain information about the Job, Setup and User which can be used in your Job's command template

Templates and Job Parsing

How Templates are Expanded

or in a Template library module. The valid *JAMS* predefined parameters are listed in Table 5–1.

The format, if specified, provides format information for the parameter's data. Valid format specifications vary depending on the parameter's data type. Valid formats for each data type are listed in Table 5–2.

Table 5–1 JAMS Predefined Parameters

Name	Description
JAMS_COMMAND_FILENAME	The full filename of the Job's command file.
JAMS_DESC_LN1	If this Job was submitted by using a Setup definition, this is the first line of the Setup's description. Otherwise, this is the first line of the Job's description.
JAMS_DESC_LN2	If this Job was submitted by using a Setup definition, this is the second line of the Setup's description. Otherwise, this is the second line of the Job's description.
JAMS_JOB_DESC_LN1	This is the first line of the Job's description.
JAMS_JOB_DESC_LN2	This is the second line of the Job's description.
JAMS_JOB_NAME	This is the name of the Job as defined in the <i>JAMS</i> database.
JAMS_LOG_DIR	The text specified in the System's Log Location field.
JAMS_LOG_SPEC	The expanded Log file specification.
JAMS_MAIL_ADR	This is the text specified in the OpenVMS Mail Address field of the Job's System Definition.
JAMS_NAME	If this Job was submitted by using a Setup definition, this is the name of the Setup. Otherwise, this is the name of the Job.
JAMS_OPER_CLASSES	This is the text specified in the Operator Classes field of the Job's System Definition.
JAMS_PRINT_DIR	This is the text specified in the Print File directory field of the Job's System definition.
JAMS_REPLY_USERS	This is the text specified in the OpenVMS Reply Username List field of the Job's System Definition.
JAMS_RON	The Job's run occurrence number. This number is unique to this specific occurrence of a Job. It is similar to the Job's queue entry number except that a RON is not reset until it exceeds 2 billion.
JAMS_SETUP_DESC_LN1	If this Job was submitted by using a Setup definition, this is the first line of the Setup's description. Otherwise, this is a null string.
JAMS_SETUP_DESC_LN2	If this Job was submitted by using a Setup definition, this is the second line of the Setup's description. Otherwise, this is a null string.
JAMS_SETUP_NAME	If this Job was submitted by using a Setup definition, this is the name of the Setup. Otherwise, this is a null string.

Templates and Job Parsing

How Templates are Expanded

Table 5–1 (Cont.) JAMS Predefined Parameters

Name	Description
JAMS_SUBMITTED_BY	This is the OpenVMS username of the person who submitted the Job. This may not be the same as the OpenVMS username which the Job runs under.
JAMS_SUBMIT_TIME	This is the date and time that the Job was submitted.
JAMS_SYSTEM_DESC	This is a description of the Job's System.
JAMS_SYSTEM_ID	This is the Job's System I.D.
JAMS_TEMP_DIR	The text specified in the Directory for Temporary files field of the Configuration screen.

Table 5–2 Data Types and Formats

Data Type	Format	Description
Text	FILL	Does not trim trailing spaces.
	integer	Expands or trims the data to <i>integer</i> bytes.
INTEGER	DEC	Base ten (decimal) notation.
	HEX	Base 16 (hexadecimal) notation.
	OCT	Base eight (octal) notation.
	integer	Base ten (decimal) notation with leading zeros to fill to <i>integer</i> bytes.
DATE	DD-MMM-YYYY	Default date format.
	MMDDYY	
	MM/DD/YY	
	MM/DD/YYYY	
	MM-DD-YY	
	MM-DD-YYYY	
	YYMMDD	
	YYYYMMDD	
TIME	HH:MM	24 hour format (military time).
	HH:MM AM	12 hour format.
	HHMM	24 hour format (military time).
	HHMMAM	12 hour format.

5.2.2 Qualified Parameter References

A Qualified Parameter Reference has the following format:

```
<<data-class\data-name\parameter-name[(format)]>>
```


Qualified Parameters are used to obtain the value of fields in a Job's Report or Parameter Definitions. The *data-class* must be one of either *REPORT*, *PRINT*, *LOGICAL* or *PARAMETER*. The *REPORT*, *PRINT* and *LOGICAL* data classes are used to obtain values from one of the Job's Report definitions and the *PARAMETER* data class is used to obtain values from one of the Job's Parameter definitions.

The *REPORT*, *PRINT* and *LOGICAL* data classes are nearly identical. In fact, in the context of a qualified parameter reference, *REPORT* and *LOGICAL* are identical. The difference between the *REPORT* and *LOGICAL* data classes is explained in Table 5–3. The *PRINT* data class is identical to the *REPORT* and *LOGICAL* data classes except that the *PRINT* data class marks the referenced Report as printed. See Section 5.2.4, for additional information on the implications of marking a Report as being printed.

The *data-name* specifies from which Report or Parameter you want to obtain a value.

The valid *parameter-names* which you can use in a Qualified Parameter Reference depend on which data class you specified. The valid parameter names for each data class are listed in Table 5–4.

The optional (*format*) specification accepts the same values as a simple parameter. These values are explained in Table 5–2.

5.2.3 Function References

A Function Reference has the following format:

```
<%function-name[(data-class[\ data-name[\ parameter-name]])]%>
```

A Function Reference expands a text module one or more times. The **function-name** is the name of the text module in the template library. The name *JAMS_JOB* is reserved and specifies the Job's command file. The text module is expanded in the context of a *data-class*. The valid data classes are explained in Table 5–3.

Table 5–3 Data Classes

Class	Description
LOGICAL	Used to specify information from a Job's Reports which have unique logical names. If a Job has five Reports defined, but two of the Reports have the same logical name, then there are only four logical name data items.
PARAMETER	Used to specify information on a Job's parameters.
PRINT	Used to specify information from a Job's Reports which will be printed and have not yet been printed. A Report is considered printed when it is referenced using the PRINT data class. A Report will not be printed if the number of copies specified in the Report definition is zero.
REPORT	Used to specify information from a Job's Reports.

Templates and Job Parsing

How Templates are Expanded

When a Function Reference is expanded, it is expanded once for each occurrence of the specified data class. If the function reference also includes the *data-name*, the function is expanded only once, for the specified data name. The *data-name* is the name of a particular occurrence of the specified *data-class*, for example, the name of a Report or Parameter. This syntax is generally used only in a Job's command template.

When a function's text module is expanded, it is done in the context of one occurrence of the specified *data-class*. When the text of the function includes a Parameter Reference, it can reference either one of the *JAMS* predefined parameters listed in Table 5–1 or a parameter specific to the data class. The parameters available for each of the data classes are listed in Table 5–4.

Table 5–4 Data Class Parameters

Class	Parameter Name	Description
LOGICAL	COPIES	The number of copies of the Report which should be printed.
	DELETE	If the retention days is zero, returns DELETE, otherwise returns NODELETE
	DESCRIPTION	The text in the Report's Description field.
	DESC_LN1	The text in the Report's Description field.
	FORM	The form on which the Report should be printed.
	FILENAME	The text in the Report's Filename field.
	LOGICAL	The Report's Logical Name.
	QUALIFIERS	The text specified in the Qualifiers field of the Report definition.
	QUEUE	The print queue where the Report should be printed.
	REPORT	The Report I.D.
	RETENTION	The number of days that the Report file should be retained.
REPORT	Same as LOGICAL	
PARAMETER	NAME	The name of the parameter.
	VALUE	The current value of the parameter.
	QVALUE	The quoted, current value of the parameter. Enclosed in double quotes if the data type is text.
PRINT	Same as LOGICAL	

5.2.4 Controlling Report Printing

It is important to understand the difference between the *REPORT*, *PRINT* and *LOGICAL* data classes and how you can use these data classes to control when and how your Reports are printed.

In the simplest case of a Job which produces one Report, you generally will not need to use the functionality provided by the three different data classes. But suppose we make the Job a little more difficult. Let's assume that the Job produces two Reports, and that we want to print the first Report before the Job finishes (while the second Report is being Generated). And, let's suppose that we want to print the second Report on two different print queues. This type of scenario occurs in most OpenVMS shops and is easily handled by *JAMS*.

5.2.4.1 Base Functionality

First let's look at base functionality. You define the Job and its two Reports. When the command file is parsed and expanded, *JAMS* will define a logical name for each of the Reports and generate two *PRINT* commands at the end of the Job.

5.2.4.2 Printing a Report More than Once

In order to make *JAMS* print the second Report twice, we simply use Job Maintenance to define a third Report. However, we give the second and third Reports the same logical name. Now when *JAMS* parses the command file it will define two logical names (because the *LOGICAL* data class treats the second and third Reports as one in the same since they have the same logical name). Then, *JAMS* will generate three *PRINT* commands at the end of the command file (because the *PRINT* data class does not merge Reports based on the logical name).

5.2.4.3 Controlling Print Time

Now, to force the first Report to print before the end of the Job, we have two options. Both options entail modifying the Job's command file so that a *PRINT* command is issued as soon as the first Report is ready to print. If we simply insert a OpenVMS *PRINT* command, the Report will certainly start printing early, but *JAMS* would issue a second *PRINT* command at the end of the Job. The trick is to reference the Report using the *PRINT* data class which will mark the Report as printed and thus prevent *JAMS* from generating a *PRINT* command at the end of the Job.

Our two options are to either use a function call to the *PRINT* function or to reference the Report using Qualified Parameter Reference and the *PRINT* data class.

5.2.5 An Example of Template Expansions

The following is an example which illustrates many of the features which have been explained in the preceding sections.

Templates and Job Parsing

How Templates are Expanded

This is an example of a Job which has two Reports. We have added one customized text module to the Template Library which is used to distribute a Report via OpenVMS Mail. This module must be invoked in the context of a REPORT, LOGICAL or PRINT data class because it references parameters which are specific to these classes. The following is the exact text of this module:

```
$!  
$!  Distribute a Report via OpenVMS Mail  
$!  
$  MAIL/SUBJECT="<<DESCRIPTION>>" -  
    /PERSONAL_NAME="JAMS Batch Job" -  
    <<LOGICAL>>  <<JAMS_SUBMITTED_BY>>  
$!
```

Note: This module is NOT included in the default Template Library, it is only shown here as an illustration.

The Job in this example runs two programs, each of which will produce one Report. The programs which the Job runs accept values for their parameters from SYS\$INPUT (as does the COBOL ACCEPT verb). The exact text of the Job's command file is shown in Example 5-2.

When this Job is requested by an end-user (via the *Job Submission* menu option), *JAMS* will parse the Job file and produce a temporary command file which is a combination of the JAMS_MAIN text module, the Job's command file and any template text modules which are referenced during the parsing. Example 5-3 shows the temporary command file which *JAMS* will create.

Templates and Job Parsing

How Templates are Expanded

Example 5–2 Sample Jobs command file

```
$!  
$RUN PRD_EXE:PROJECT_DETAIL.EXE  
<<JAMS_JOB_NAME>>1  
<<REPORT\PROJDTL\DESCRIPTION>>2  
<<START_DATE (YYMMDD) >>3  
<<END_DATE (YYMMDD) >>  
<<PROJECT>>  
$!  
<%PRINT (PRINT\PROJDTL) %>4  
$!  
$RUN PRD_EXE:PROJECT_SUMMARY.EXE  
<<JAMS_JOB_NAME>>  
<<REPORT\PROJSUM\DESCRIPTION>>  
<<START_DATE (YYMMDD) >>  
<<END_DATE (YYMMDD) >>  
<<PROJECT>>  
$!  
<%MAIL (PRINT\PROJSUM) %>5  
$!
```

- 1** This program accepts the Job Name so it can be printed in the headings of the report. (A simple parameter reference to a *JAMS* predefined parameter).
- 2** It also accepts the Report Description which will be printed in the headings of the report. (A qualified parameter reference to a Report data class parameter).
- 3** Accept the date parameters and force the correct format. (A simple parameter reference with formatting).
- 4** Tell JAMS to insert the PRINT command here so that the report will start printing before the Job has finished. (A function reference).
- 5** Invoke our custom MAIL function to mail the second report to the user who submitted the Job.

Example 5–3 Sample Job's command file after parsing

```
$!  
$! JAMS Main Job Structure 1  
$!  
$ JAMS_RON := 834962  
$ JAMS_JOB_NAME := PROJECT_DTL_SUM  
$ JAMS_SETUP_NAME :=  
$ JAMS_SYSTEM_ID := PROJECTCTL  
$ JAMS_SUBMITTED_BY := GBUSH  
$ JAMS_SUBMIT_TIME := 15-APR-1991 09:53  
$!  
$ START_DATE == "02-AUG-1991" 2  
$ END_DATE == "11-APR-1991"
```

Example 5–3 Cont'd on next page

Templates and Job Parsing

How Templates are Expanded

Example 5–3 (Cont.) Sample Job's command file after parsing

```
$!  
$ DEFINE PROJDTL - 3  
    'f$parse("", "PROJCTL_REP:PROJDTL.000CBD92",,, "SYNTAX_ONLY")  
$ DEFINE PROJSUM  
    'f$parse("", "PROJCTL_REP:PROJSUM.000CBD92",,, "SYNTAX_ONLY")  
$!  
$!  
$RUN PRD_EXE:PROJECT_DETAIL.EXE 4  
PROJECT_DTL_SUM  
Detailed Expenses for a Project  
900802  
910411  
DSTORM  
$!  
$ SET FILE/EXPIRATION_DATE="TODAY+5-" PROJSUM 5  
$ PRINT/NOFEED/QUEUE=OVAL_LN03/NODELETE -  
    /FORM=DEFAULT/COPIES=1 -  
    PROJSUM  
  
$!  
$!  
$RUN PRD_EXE:PROJECT_SUMMARY.EXE  
PROJECT_DTL_SUM  
Summary of Expenses for a Project  
900802  
910204  
DSTORM  
$!  
$!  
$! Distribute a Report via OpenVMS Mail 6  
$!  
$ MAIL/SUBJECT="Summary of Expenses for a Project" -  
    /PERSONAL_NAME="JAMS Batch Job" -  
    PROJSUM GBUSH  
  
$!  
$!7  
$!
```

- 1 Define DCL symbols for some of the predefined parameters.
 - 2 Define DCL symbols for each of the Job's parameters.
 - 3 Define logical names for our report files.
 - 4 This is the beginning of our Job's command file.
 - 5 This is the text from the PRINT function which we specified in our command file.
 - 6 Here is our expanded MAIL function.
 - 7 The JAMS_MAIN text module calls the PRINT function at the end of the JOB, but since we have already referenced both of our reports using the PRINT data class, there are no reports to print.
-

5.2.6 Trapping DCL errors

When you write DCL command procedures you can use the `ON ERROR` statement to trap errors which occur during the processing of the command procedure. The standard *JAMS* template library does not include any error trapping. If an error occurs the command exits and returns a status value which represents the error which caused the job to fail. *JAMS* uses the exit status of a job to determine if the job was successful.

If you want to trap errors in your command procedures, make sure that you save the status code of the error which caused the job to fail and use this saved value when the job exits. This is the only way that *JAMS* will know that the job failed.

Example 5–4 shows a fragment of a command procedure which traps errors and also exits with the correct status code.

Example 5–4 Trapping DCL errors

```
$!  
$ ON ERROR THEN GOTO ERROR_RTN      ! Set error trapping  
$!  
.  
.  
.  
$!  
$ERROR_RTN:  
$    saved_status = $STATUS          ! Save the bad status value  
$!  
$!    Cleanup after the error  
$!  
$    EXIT saved_status              ! Exit with the bad status value  
$!
```

A

Error Messages in order by Identifier

This appendix provides a description of the error messages which can be displayed or returned by the *JAMS* software system.

A.1 Error Message Format

JAMS uses standard VMS messages to return information to the user. Standard VMS messages have the following general format:

```
%FACILITY-L-IDENT, TEXT
```

FACILITY

A VMS facility or component name. *JAMS* messages use the facility name *JAMS*. *JAMS* may also display messages which are returned by other facilities. A percent sign (%) prefixes the first message in a sequence of related messages. Subsequent messages in the sequence are prefixed with a hyphen (-).

L

An indicator of the severity of the error. The possible values and their meaning are listed in the following table:

Code	Meaning
S	Success
I	Informational
W	Warning
E	Error
F	Fatal

IDENT

An abbreviation of the message.

TEXT

The text of the error message.

Note: The messages in this appendix are arranged in alphabetical order based on the identifier of the message. If you want to locate a message based on the message's text, you can look up the identifier in Appendix B.

Error Messages in order by Identifier

Error Message Format

ABNTERM, Message from the JAMS Batch Job Monitor at dd-mmm-yyyy
Abnormal termination of job xxxx, final status was...

Explanation: This message is broadcast to your terminal is a batch job terminates abnormally and you are on the notification list.

User Action: This is notification that one of your jobs died, you may want to find out why.

ALRAUTOSUBMIT, Job has already been auto-submitted after this time.

Explanation: The JAMS_AUTOSUBMIT Job did not submit this Job because the Job has already been submitted with a scheduled time which is on or after the last time scheduled date/time for this Setup. This can happen if you re-run JAMS_AUTOSUBMIT or if you change the scheduled time and/or parameter values for JAMS_AUTOSUBMIT.

User Action: You should know what has caused this overlap of scheduled times. If you are re-running JAMS_AUTOSUBMIT because of a system crash, double check the last job in the report which has this message and make sure that it was in fact submitted.

AMBIGPERIOD, Ambiguous period specified.

Explanation: In a text date specification, you referenced a specific period which is defined for more than one date type. This frequently happens when your company uses more than one fiscal year and you define both fiscal year date types using the same specific names for each month.

User Action: You can change your date type definitions and specify unique names for the specific periods however, since these date types are already in use, this change may affect other people and/or Job definitions.

You could also use both the specific period and date type in your date specification such as "FIRST DAY OF fiscal period_01" instead of "FIRST DAY OF period_01".

ASTLMREACHED, AST Limit reached, restarting with a larger ASTLM.

Explanation: The JAMS_SCHEDULE process nearly exceeded its ASTLM quota. It is restarting with a larger ASTLM quota.

User Action: None.

CANCELJOB, Job Canceled by precheck Job.

Explanation: If a Precheck Job exits with this status, the Job which caused the Precheck to run will be deleted from the batch queues.

User Action: None.

CANTSUBMIT, Errors encountered while trying to submit job.

Explanation: One or more errors were encountered while submitting a Job. Subsequent messages will describe the errors which were encountered.

User Action: Review the subsequent error messages and take appropriate action.

Error Messages in order by Identifier

Error Message Format

CHARDEFINED, Defined characteristic xxxxxx, number nn..

Explanation: When the Schedule process is starting, it checks for a number of queue characteristics. If these characteristics are not defined, it will define them and issue this message.

User Action: None.

CHARERROR, Unable to obtain or define a JAMS characteristic.

Explanation: A *JAMS* procedure could not obtain or define the value of one of the JAMS characteristics. *JAMS* characteristics are defined by the JAMS_SCHEDULE process when it starts up.

User Action: Use the SHOW/QUE/CHAR command to display the queue characteristics which are defined on your system. If there are no available queue characteristic numbers, *JAMS* cannot define a new characteristic. If one of the *JAMS* characteristics is not defined and there are available characteristic numbers, make sure that the VMS Queue manager is running and then use the STOP SCHEDULE and START SCHEDULE commands to restart the JAMS_SCHEDULE process.

CHECKJOBS, Discovered missing job on dd-mmm-yyyy.

Explanation: A batch job which was executing seems to have disappeared. If the monitor cannot find the job in a reasonable period of time, it will declare the job missing in action.

User Action: If the job is subsequently declared MIA, submit an SPR.

CIRCPRECHK, This Pre-Check would create a circular reference.

Explanation: The Job you specified as a Pre-check Job would create a circular list of Pre-checks. A circular list of Pre-checks is one which forms a circle of Jobs. For example, JOBA has a Pre-check of JOBB which has a Pre-check of JOBC which has a Pre-check of JOBA. If you were allowed to form a circle like this, when you submitted one of the Jobs in the circle, *JAMS* would continuously submit the circle of Jobs.

User Action: None.

CONFDEL, Press Gold/D to confirm deletion.

Explanation: You are in delete mode and you have selected an item to be deleted. The item which you have selected is displayed for your review.

User Action: If the item which is displayed is the one which you want to delete, press Gold/D to confirm the deletion, otherwise, press any other key to abort the deletion.

CRENETWORK, Creating a NETWORK process on !%D.

Explanation: This message logs when a Monitor process tries to start a new JAMS_NETWORK process.

User Action: None.

Error Messages in order by Identifier

Error Message Format

CRESCHED, Creating a SCHEDULE process on dd-mmm-yyyy.

Explanation: This message may appear in a Monitor processes log file. It is used to provide a audit trail of the creation of Schedule processes.

User Action: None.

CURRENTINV, /CURRENT is invalid for this data type.

Explanation: The /CURRENT qualifier (or option bit) is valid only for Variables with a data type of DATE, TIME or DATETIME.

User Action: Correct your code or command or change the Variables data type.

CWSHUT, Clusterwide JAMS shutdown requested.

Explanation: This is an informational message. When you use the STOP MONITOR/CLUSTER command to shutdown the monitor and schedule processes on the entire cluster, this message is displayed.

User Action: None.

DATANOTAVAIL, Specified data is not available.

Explanation: JAMS_GET_DATA cannot return the specified data because it is not available.

User Action: None.

DATEDNOTFOU, Unable to locate record with this Type and date.

Explanation: You are trying to view or modify a date definition and a record with the specified Date Type and date cannot be found.

User Action: Verify the data entered, use the Find key to display a list of dates.

DATENOTFOU, Date Type not found.

Explanation: You are trying to use a Date Type which is not defined.

User Action: Enter a valid Date Type or, use the Find key to list the valid Date Types.

DATENSU, Dates for this period have not been defined.

Explanation: You have entered an English language date specification which references a valid, user defined date type but, the dates for the time period you are referencing have not been defined. For example, the date specification "1ST WORKDAY OF NEXT FISCAL NOVEMBER" may be a valid date specification but, if the start of Fiscal November of next year has not been defined, this message will be returned.

User Action: Verify that your date specification is correct. If so, then you need to define the specific dates which you are referring to.

Error Messages in order by Identifier

Error Message Format

DATEONFILE, This Date Type is already on file.

Explanation: You are in Add mode and the Date Type which you entered is already defined.

User Action: Select a different Date Type or use Modify mode.

DUPPARAM, Duplicate Parameter names on one Job are not allowed.

Explanation: You tried to define a parameter which has the same name as another parameter in this job. Parameter names must be unique within a Job.

User Action: Select a different parameter name.

DUPREPORT, Duplicate Report names on one Job are not allowed.

Explanation: You tried to define a report which has the same name as another report in this job. Report names must be unique within a Job.

User Action: Select a different report name.

DUPTRGACTION, This Trigger already contains this Action.

Explanation: A Trigger cannot submit the same Job more than once.

User Action: You could create a duplicate Trigger or create a Job which contains *JAMS* submit commands to submit the same Job more than once.

DUPTRGEVENT, This Trigger already contains this Event.

Explanation: A Trigger cannot reference the same Job of Variable in two events.

User Action: If you are trying to define a Trigger with OR logic, create two separate Triggers.

ENDBEFORESTART, The end time cannot be before the scheduled time.

Explanation: The end time of a Setup which resubmits itself must be after the Setups schedule time.

User Action: If you want to have a Job resubmitted during a time span which includes midnight, you must define two Setups. One for the range before midnight and one for the range after midnight.

EXECUTING, Job should be executing.

Explanation: This is the message code is used on job history records during the time that the job is executing.

User Action: If the job is actually executing, there is no action. If the job is not executing, please submit an SPR.

FIELDFULL, Field is full !

Facility: SCU, Screen Control Utility

Explanation: You are trying to enter more characters into a field than the field definition allows.

User Action: None.

Error Messages in order by Identifier

Error Message Format

FILEERR, Unable to locate/access job file xxxxxx

Explanation: While trying to submit a job, the *JAMS* could not locate or open the command file specified in the Job definition.

User Action: Verify your Job and System definitions. The name of the file is specified in the Job definition. The device and directory are specified in the System definition. Any logical names referenced in the file specification must be defined with the /EXECUTIVE qualifier.

GOLD, Gold...

Facility: SCU, Screen Control Utility

Explanation: You pressed the Gold key (PF1) and the system is waiting for you to press another key to complete the Gold key sequence.

User Action: If you did not intend to press the Gold key, press the Gold key again to cancel the sequence.

INCDECRINV, /INCREMENT and /DECREMENT are invalid for this data type.

Explanation: The /INCREMENT or /DECREMENT qualifier (or option bit) is not valid for the Variables data type.

User Action: Correct your code or command or change the Variables data type.

INPUTREQ, Input is required.

Facility: SCU, Screen Control Utility

Explanation: You left a field blank and the field is a required entry.

User Action: Enter a value for the field or, press the Exit key to abort the entry.

INTERR, Internal error, please submit an SPR.

Explanation: This error should never be encountered.

User Action: Please submit an SPR.

INVACK, Invalid ACK, serial number !XL, status !XL, NWIP !UL

Explanation: The message logs an invalid ACK from a remote network process. The message will be resent.

User Action: If this is a common occurrence, please submit an SPR.

INVANSWER, Invalid answer, reply YES, NO, Y or N.

Explanation: You entered an invalid answer to a confirmation prompt.

User Action: Respond with YES, NO, Y or N.

INVBATCHQ, Invalid batch queue specified.

Explanation: The specified queue is not a valid batch queue.

User Action: Enter a valid batch queue.

Error Messages in order by Identifier

Error Message Format

INVBOOLEAN, Invalid boolean value, enter T, F, Y, N, 0 or 1.

Explanation: You specified a value for a boolean Variable which is invalid.

User Action: Correct your code or command or change the Variables data type.

INVCHAR, Invalid Character.

Facility: SCU, Screen Control Utility

Explanation: You typed a character which is invalid for the current field, such as typing a letter into a numeric field.

User Action: None.

INVCONDITION, Invalid condition code, use the FIND key.

Explanation: You entered an invalid condition.

User Action: Use the Find key to list all of the valid conditions.

INVCTX, Invalid context passed in.

Explanation: You called *JAMS_GET_DATA* but the context you passed is not valid.

User Action: You must pass a valid context to *JAMS_GET_DATA*.

INVDATAID, Invalid data identifier.

Explanation: You called *JAMS_GET_DATA* but one of the data identifiers which you passed is not valid.

User Action: You must pass valid data identifiers to *JAMS_GET_DATA*.

INVDATATYPE, Supported datatypes are TEXT, INTEGER, DATE and TIME.

Explanation: The data type for parameters must be TEXT, INTEGER, DATE or TIME.

User Action: Enter a valid data type.

INVDATE, Invalid date.

Explanation: The date entered is not valid.

User Action: Enter a valid date in the form MM/DD/YY.

INVFORMAT, Invalid format.

Explanation: You entered a format which is not valid for this parameters data type. Clear the field or press the Find key to list the valid formats.

User Action: Enter a valid format.

Error Messages in order by Identifier

Error Message Format

INVFUNC, Invalid function detected during parsing of job.

Explanation: While parsing a job, *JAMS* found a function reference which does not exist in the System's template library.

User Action: Correct the job file or add the text module to the template library.

INVIL, Invalid Item List.

Explanation: The item list passed to the *JAMS* routine is not valid or is missing required information.

User Action: Correct your program.

INVLENGTH, The length must be between 1 and 80.

Explanation: The length of a text parameter must be between one and 80 characters long.

User Action: Enter a number between 1 and 80.

INVMBXTYPE, Invalid message type of x received in the registration mailbox.

Explanation: An invalid message was placed in the *JAMS* monitor mailbox.

User Action: Please submit an SPR.

INVPARAM, Invalid Parameter name.

Explanation: Parameter names must be valid VMS symbol names. They also must not begin with the text "JAMS_".

User Action: Enter a valid parameter name.

INVPARSE, This entry must be Y (Yes), N (No) or J (Jacket).

Explanation: You entered a character other than Y, N or J.

User Action: Correct your entry.

INVPERIOD, Invalid period.

Explanation: The date period specified is not valid.

User Action: Select a valid period.

INVPRINTQ, Invalid print queue specified.

Explanation: The specified queue is not a valid print queue.

User Action: Select a valid print queue.

INVRETDSC, Invalid return descriptor.

Explanation: You called *JAMS_GET_DATA* but one of the return descriptors which you passed is not valid.

User Action: You must pass valid return descriptors to *JAMS_GET_DATA*. Also, the data type of the return descriptor must be compatible with the data type of the data identifier.

Error Messages in order by Identifier

Error Message Format

INVSCHDATE, Invalid schedule date.

Explanation: The date specified is not a valid date specification.

User Action: Enter a valid date specification.

INVSEVERITY, Invalid severity code, must be S, I, W, E or F.

Explanation: The severity code must be S (Success), I (Informational), W (Warning), E (Error) or F (Fatal).

User Action: Enter a valid severity code.

INVSPECIFIC, Not valid for this Date Type, use the Find Key.

Explanation: You entered a Specific date type which is not valid for this date.

User Action: Press the Find key to list the valid Specific Date Types.

INVSYMNAME, Name must start with A-Z, contain A-Z, 0-9, \$ and underscore.

Explanation: This name must be a valid VMS symbol name. This means that it must begin with an alphabetic character and contain only the characters A-Z, 0-9, \$ (dollar sign), and _ (underscore). Also the last character may not be an underscore.

User Action: Enter a valid name.

INVTEXT, Invalid characters in text value

Explanation: A Variable with the data type of TEXT can only contain printable ASCII characters. Any characters outside the range of ASCII 32 through 127 will cause this error.

User Action:

INVUSERNAME, This is not a valid VMS username, not found in SYSUAF.

Explanation: You entered a VMS username which could not be found in the default system authorization file (SYSUAF). If you want to specify a username in a System definition, you must first create the username in the SYSUAF file.

User Action: Select a different username, leave the username field blank, or add the desired username to the SYSUAF file.

INVWORKDAY, Non-workday processing must be S, I, or D. (Schedule, Ignore, Defer)

Explanation: You tried to enter an invalid value into the non-workday processing field.

User Action: The valid values are S (Schedule the job even on non-workdays), I (Ignore the job on non-workdays) or D (Defer the job until the next workday).

Error Messages in order by Identifier

Error Message Format

ISFMEM, Insufficient memory for allocation of dynamic data areas.

Explanation: *JAMS* could not allocate additional virtual memory.

User Action: A user or system quota has been exceeded, check the PGFLQUO value in the users UAF record, also check the SYSGEN parameter VIRTUALPAGECNT and make sure that there is adequate page file space.

JCQREADERR, Unable to read JCQ record ## at dd-mmm-yyyy

Explanation: A record in the JCQ file could not be read. The RMS status value should follow this error and provide additional information about why the read failed.

User Action: Please submit an SPR.

JCQUPDERR, Unable to update JCQ header record at dd-mmm-yyyy

Explanation: The header record in the JCQ file could not be updated. The RMS status value should follow this error and provide additional information about why the update failed.

User Action: Please submit an SPR.

JCQWRITERR, Unable to write JCQ record ## at dd-mmm-yyyy

Explanation: A record in the JCQ file could not be written. The RMS status value should follow this error and provide additional information about why the write failed.

User Action: Please submit an SPR.

JOBNIIL, Job or Setup not specified in Item List.

Explanation: The item list which you passed did not contain either a Job name or a Setup name.

User Action: Correct your program.

JOBNOTFOU, Job not found.

Explanation: The specified job was not found in the *JAMS* database.

User Action: Enter a valid Job Name.

JOBONFILE, This Job is already on file.

Explanation: You are in Add mode and the Job Name which you entered is already defined.

User Action: Select a different Job Name or use Modify mode.

JOBQUEERROR, Fatal VMS queue manager error at dd-mmm-yyyy hh:mm

Explanation: The VMS queue manager returned an error which *JAMS* regards as fatal. The JAMS_SCHEDULE process will shutdown so that it can restart on a node with a working queue manager.

User Action: Check and correct your VMS queue manager.

Error Messages in order by Identifier

Error Message Format

JOBSINSYS, There are still Jobs with this System I.D., deletion not allowed.

Explanation: You cannot delete a System which still has Jobs, Setups or Menus which refer to it.

User Action: Delete all references to the System before you delete the System.

LINKERROR, Link error to node !AS

Explanation: A DECnet link to a remote node encountered an error. The status which caused the error is also displayed.

User Action: If the status which caused the error is not explainable, please submit an SPR.

LINKFAILURE, Link to node !AS failed

Explanation: A DECnet link to a remote node failed. The status which caused the failure is also displayed.

User Action: If the status which caused the failure is not explainable, please submit an SPR.

LINKNAK, Link replied with NAK

Explanation: A remote node replied with a NAK. The message will be resent.

User Action: If this is a common occurrence, please submit an SPR.

LISTTRUNC, List has been truncated, too much data found to list it all.

Explanation: JAMS list screens have a limit of 64K bytes. If this limit is exceeded, the list is truncated so some of the requested information is not displayed.

User Action: Enter more restrictive selection criteria.

MBXERR, Error accessing monitor mailbox.

Explanation: The JAMS_REGISTRAR program encountered an error while trying to open the mailbox to the Monitor process.

User Action: Please submit an SPR.

MENUNOTFOU, Menu not found.

Explanation: The specified Menu is not in the JAMS database.

User Action: Enter a valid Menu name.

MENUONFILE, This Menu is already on file.

Explanation: You are in Add mode and the Menu Name which you entered is already defined.

User Action: Select a different Menu Name or use Modify mode.

Error Messages in order by Identifier

Error Message Format

MIAJOB, Declared Job MIA on dd-mmm-yyyy.

Explanation: A batch job which was being monitored by *JAMS* has disappeared. When the Monitor process discovers a missing job, it uses this message to record this abnormal event in the log file.

User Action: This should never happen. If it does, please submit an SPR.

MJALRCOMPLETED, This Job has already completed.

Explanation: You tried to modify a job and it has already completed.

User Action: None.

MJALRRUNNING, This Job is already running.

Explanation: You tried to release or reschedule a job and it is already executing.

User Action: None.

MJNOJOBS, No jobs selected for display.

Explanation: There are no jobs which meet your selection criteria. The Job monitor will re-display this message every 15-20 seconds until there are jobs which should be displayed.

User Action: None.

MJNOTRUNNING, This Job is not running.

Explanation: You tried to requeue a job and it is not executing.

User Action: None.

MONALRSTART, The JAMS monitor has already been started.

Explanation: You tried to start the *JAMS* monitor when it was already running.

User Action: Don't do this.

MONSHTDWN, MONITOR shutting down on dd-mmm-yyyy.

Explanation: A monitor shutdown was requested.

User Action: None.

MONSHTREQ, MONITOR shutdown requested on xxxxxx.

Explanation: A shutdown was requested, the monitor is shutting down.

User Action: None.

Error Messages in order by Identifier

Error Message Format

MONSHTRUN, MONITOR shutdown requested on xxxx with ## jobs still executing.

Explanation: A shutdown was requested, the monitor is shutting down. However, there were still batch jobs running whose completion will not be monitored.

User Action: If you want to maintain accurate execution history, wait until all batch jobs have completed before you shutdown the *JAMS* monitor.

MONSTART, MONITOR starting up on dd-mmm-yyyy.

Explanation: Logs the date and time that the monitor started.

User Action: None.

MUSTBEBW, This entry must be either B (Better) or W (Worse).

Explanation: You entered a character other than B or W.

User Action: Correct your entry.

MUSTBEYN, This entry must be either Y (Yes) or N (No).

Explanation: You must enter a Y or a N. In some cases, you may leave the field blank.

User Action: Enter a Y, N or blank.

MUSTFILL, This field must be filled.

Facility: SCU, Screen Control Utility

Explanation: You entered a value into a field which did not completely fill the field and the field must be completely filled.

User Action: Correct your entry or, blank the field.

NETNOTRESP, NETWORK process was started but is not responding.

Explanation: The JAMS_MONITOR process is trying to start a network process but it does not seem to be working.

User Action: Please submit an SPR.

NETNOTRUN, NETWORK process is not running.

Explanation: You issued the STOP NETWORK command and there is not a network process running.

User Action: None.

NETNOTRUNNING, The JAMS NETWORK process may not be running.

Explanation: There is work for the network process but it is not running.

User Action: If the network process is in fact running, please submit an SPR.

Error Messages in order by Identifier

Error Message Format

NETSTART, NETWORK process is starting up.

Explanation: This message logs the start of a net JAMS_NETWORK process.

User Action: None.

NEWLOG, Creating new JAMS.LOG log file in JAMS_DATA:

Explanation: This message is issued in response to the RENEW LOG_FILE command.

User Action: None.

NOCHARAVAIL, No available characteristic numbers.

Explanation: *JAMS* needs to use VMS queue characteristics to control the execution of batch jobs. When the JAMS_SCHEDULE process starts up, it checks for the *JAMS* characteristics and will define them if they are not found.

This message is issued if *JAMS* cannot define the characteristics because there are no available characteristic numbers.

User Action: Review your VMS queue characteristics and delete at least one of them.

NODEGNOTFOU, Node Group not found.

Explanation: The specified Node Group was not found in the *JAMS* database.

User Action: Enter a valid Node Group Name or, use the Find key.

NODEGONFILE, This Node Group is already on file.

Explanation: You are in Add mode and the specified Node Group is already on file.

User Action: Select a different Node Group name or use Modify mode.

NODENOTFOU, Node Definition not found.

Explanation: The specified Node Definition was not found in the *JAMS* database.

User Action:

NODEONFILE, This Node Definition is already on file.

Explanation:

User Action: Enter a valid Node Definition name or, use the Find key.

NODESHUT, JAMS shutdown requested.

Explanation: You issued the STOP MONITOR command.

User Action: None.

Error Messages in order by Identifier

Error Message Format

NODOCLOCATION, The Documentation Location is not specified.

Explanation: You must specify the Documentation Location in the System Definitions before you can create documentation for a System or any of the Job's in the System.

User Action: Define the Documentation Location in the System Definition.

NOFIND, Sorry, FIND is not available for this field.

Facility: SCU, Screen Control Utility

Explanation: You pressed the FIND key while the cursor was in a field which does not have a Search routine associated with it.

User Action: None.

NOHELP, Sorry, help is not available for this field.

Facility: SCU, Screen Control Utility

Explanation: You pressed the Help key while the cursor was in a field which does not have any help text associated with it.

User Action: Press the Help key again to obtain full screen help.

NOHISTORY, No History records found which match the selection criteria.

Explanation: This is self explanatory.

User Action: Modify your selection criteria. Remember that the Job Name uses VMS wildcards. Entering "ABC" will list only the Job "ABC". If you want to list all Jobs which start with "ABC", then enter "ABC*".

Also remember that the date range applies to the Jobs submit date.

NOJCQHEAD, Unable to read JCQ header record at dd-mmm-yyyy hh:mm.

Explanation: The JCQ file is corrupt.

User Action: Please submit an SPR. Perform the following steps to recover your JCQ.DAT file.

- 1 Stop the Monitor and Schedule processes on all nodes in the VAXcluster.
- 2 Delete all versions of the JCQ.DAT file in the directory pointed to by the logical name JAMS_DATA.
- 3 Restart the Monitor and Schedule processes.

NOJOBS, No Jobs found which match the selection criteria.

Explanation: This is self explanatory.

User Action: Modify your selection criteria. Remember that the Job Name uses VMS wildcards. Entering "ABC" will list only the Job "ABC". If you want to list all Jobs which start with "ABC", then enter "ABC*".

Error Messages in order by Identifier

Error Message Format

NOMENUENTRIES, There are no entries in this menu.

Explanation: You selected a Menu line item but, the menu built from the selected menu definition did not have any entries.

User Action: Correct the menu definition.

NOMENUS, No Menus found which match the selection criteria.

Explanation: This is self explanatory.

User Action: Modify your selection criteria.

NONWIPHEAD, Unable to read NWIP header record at !%D.

Explanation: The NWIP file may be corrupt.

User Action: Please submit an SPR.

NOPARAMS, There are no Parameters defined for this Job or Job Setup.

Explanation: You pressed Gold/P but this Job or Setup does not have any Parameters.

User Action: None.

NOREPORTS, There are no Reports defined for this Job.

Explanation: You pressed Gold/R but this Job does not have any Reports.

User Action: None.

NOSETUPS, No Setups found which match the selection criteria.

Explanation: This is self explanatory.

User Action: Modify your selection criteria.

NOSPECIFIC, This Date Type does not have any Specific Types.

Explanation: You pressed the Find key to list the valid Specific Date Types for the current Date Type but this Date Type does not have any valid Specific types.

User Action: You are not allowed to enter a Specific Date Type.

NOTIMPLEMENT, Function is not implemented

Explanation: This message code is returned when you call an entry point in the *JAMS* sharable image which has not been implemented.

User Action: If this message is returned by a *JAMS* program, please submit an SPR. If this message is returned to user-written code, please check your code. You can call our technical support line if you cannot determine the cause of the problem.

NOTRGS, No Triggers found which match the selection criteria.

Explanation: There are no Triggers whose name matches the wildcard expression which you entered.

User Action: Change your wildcard expression or add the Trigger.

Error Messages in order by Identifier

Error Message Format

NOVARS, No Variables found which match the selection criteria.

Explanation: There are no Variables whose name matches the wildcard expression which you entered.

User Action: Change your wildcard expression or add the Variable.

NWIPREADERR, Unable to read NWIP record !ZL at !%D

Explanation: The NWIP file may be corrupt.

User Action: Please submit an SPR.

NWIPUPDERR, Unable to update NWIP header record at !%D

Explanation: The NWIP file may be corrupt.

User Action: Please submit an SPR.

NWIPWRITERR, Unable to write NWIP record !ZL at !%D

Explanation: The NWIP file may be corrupt.

User Action: Please submit an SPR.

OBSFILSTR, Obsolete file structures, please refer to the Installation Guide.

Explanation: There is a mismatch between your *JAMS* software and data file structure. This may happen when you install a new version of *JAMS* and you do not upgrade all of your *JAMS* data files.

User Action: Refer to the installation instructions for information on how to upgrade your data files. Generally this is performed by the installation procedure. If the installation procedure did not upgrade your *JAMS* database, the command procedure *JAMS_CONVERT_DATABASE* in the *SYS\$UPDATE* directory can be used to to this.

OPENERR, Unable to open *JAMS* database file xxxxxx.

Explanation: *JAMS* could not locate, or could not open, one of the files which makes up the *JAMS* database.

User Action: You will see this problem when the *JAMS* start-up file (*JAMS_STARTUP.COM*) has not been executed. This procedure defines *JAMS* logical names and installs images with the privileges they will need to open the *JAMS* database.

OPENJCQ, Unable to open JCQ file.

Explanation: The JOB Completion Queue file (*JCQ.DAT*) could not be opened.

User Action: Make sure that the *JAMS* start-up file (*JAMS_STARTUP.COM*) has been executed.

OPENNWIP, Unable to open NWIP file.

Explanation: The NWIP file cannot be opened.

User Action: Please submit an SPR.

Error Messages in order by Identifier

Error Message Format

PARSEERR, Error while parsing the Job file.

Explanation: *JAMS* could not parse the Job's command file. This message should be followed by additional messages which will provide more detailed information about why the command file could not be parsed. Usually, this is caused by an error in the command file or in a template customization.

User Action: Review subsequent messages and correct the problem.

PENDING, Job was submitted but has not started yet.

Explanation: This message code is used on job history records between the time that a job is submitted and the time that it actually starts executing.

User Action: None.

PRECHKSAME, The pre-check job cannot be the same as the job itself.

Explanation: A Job cannot have itself as a Precheck Job. This would cause in endless loop because a Precheck Job could also have a Precheck job.

User Action: Correct your entry.

PRESSKEY, Press any key to continue.

Facility: SCU, Screen Control Utility

Explanation: *JAMS* is displaying some information and waiting for you to confirm receipt of the information.

User Action: "Press any key" actually means almost any key. Depending on the application, there may be active function keys.

RCVUNREGTRM, Received a termination message for an unregistered job.

Explanation: The *JAMS_SCHEDULE* process received a termination message for a job which it did not know about.

User Action: Please submit an SPR.

RECCNG, Another user has changed this record while you were working.

Explanation: When you are using the *JAMS_MASTER* sub-system to update the *JAMS* database, the data you are working with is *not* locked while it is displayed on your terminal. After you have made your changes and pressed Return, the data is re-read, and a time stamp is compared to the time stamp when you initially read the data. If these two times are different it means that someone else has modified the exact same data you were working with between the time it was displayed on your terminal and the time you pressed Return.

This approach to record locking reduces locking problems but results in the possibility of this problem occurring.

User Action: When this error is detected, the data you were working with is re-read and displayed on the terminal. You may notice the changes

Error Messages in order by Identifier

Error Message Format

made by the other user. You should review the data, re-enter your changes and press Return.

REDIRERROR, Error encountered when redirecting a network request.

Explanation: When a JAMS_MONITOR process was trying to redirect a connection request to the node in the VAXcluster which is running the JAMS_NETWORK process, an error was encountered. The status which caused the error is also displayed.

User Action: Please submit an SPR.

SCHMSGMAX, Too many messages stacked up.

Explanation: The messages which the JAMS_SCHEDULE process sends out to Job Monitors were stacking up. The JAMS_SCHEDULE process will restart.

User Action: Please submit an SPR.

SCHNOTRESP, SCHEDULE process was started but is not responding.

Explanation: The JAMS Schedule process has been started but is not responding. This indicates a serious problem.

User Action: Check the Schedule log file for messages which will help determine the reason for the failure of the Schedule process. The name of the log file is JAMS_DATA:SCHEDULE.LOG. If you cannot determine the cause of the schedule process failures, please submit an SPR.

SCHNOTRUN, SCHEDULE process is not running.

Explanation: You issued the STOP SCHEDULE command and there is not a schedule process running.

User Action: None.

SCHNOTRUNNING, The JAMS SCHEDULE process may not be running.

Explanation: The JAMS Schedule process is not doing its job, even though the process exists. This is not a normal condition, if it persists or occurs frequently, please submit an SPR.

User Action: Make sure that the JAMS schedule process is indeed running. Most of the time, this message is caused because the node which is running the schedule process has suspended operations or is hung.

If the node running the Schedule process is not hung, try stopping and restarting the Schedule process with the STOP SCHEDULE and START SCHEDULE commands.

If this does not solve the problem, stop the Schedule process and check the Schedule log file for messages which will help determine the reason for the failure of the Schedule process. The name of the log file is JAMS_DATA:SCHEDULE.LOG. If you cannot determine the cause of the schedule process failures, please submit an SPR.

Error Messages in order by Identifier

Error Message Format

SCHSHTDWN, SCHEDULE process shutting down on dd-mmm-yyyy.

Explanation: Logs the date and time that the Schedule process shut down.

User Action: None.

SCHSHTREQ, SCHEDULE process on !AS has been signaled to shutdown.

Explanation: Reply from the STOP SCHEDULE command.

User Action: None.

SCHSLOW, The JAMS SCHEDULE process seems slow.

Explanation: The *JAMS* Schedule process is not doing it's job, even though the process exists. This is not a normal condition, if it persists or occurs frequently, please submit an SPR.

User Action: Make sure that the *JAMS* schedule process is indeed running. Most of the time, this message is caused because the node which is running the schedule process has a very heavy workload and the schedule process is indeed running slowly.

SCHSTART, SCHEDULE process starting up on dd-mmm-yyyy.

Explanation: Logs the date and time that the Schedule process started.

User Action: None.

SELFDEPEND, A Job may not depend upon itself.

Explanation: A Job may not depend upon it's own completion. If a Job did depend upon it's own completion and the Job ever failed, a subsequent run would remain in a pending state until manually released.

User Action: Dependencies must refer to a Job other than the one which the dependency is being defined for.

SETUPNOTFOU, Setup not found.

Explanation: The specified Setup is not in the *JAMS* database.

User Action: Enter a valid Setup.

SETUPONFILE, This Setup is already on file.

Explanation: You are in Add mode and the Setup Name which you entered is already defined.

User Action: Select a different Setup Name or use Modify mode.

SINCEDPND, The Since and Depend Job may not be the same Job.

Explanation: The Since Job and Dependent Job may not be the same because one a Job has completed twice, it has always completed since the last time it ran.

User Action: Choose a different Job for either the Since or the Dependent Job.

Error Messages in order by Identifier

Error Message Format

SPECINUSE, You deleted a Specific Type which is still in use !!

Explanation: While modifying a Date Type definition, you deleted a Specific Type which still has Dates defined for it.

User Action: Delete all Dates which reference the Specific Type before you delete the Specific Type.

STILLDATES, You must delete all of the dates of this type before deleting the type.

Explanation: You tried to delete a Date Type definition which still has Dates defined for it.

User Action: Delete all Dates which reference a Date Type before you delete the Date Type.

STILLDPND, There are still other Jobs with Dependencies which refer to this Job.

Explanation: You cannot delete a Job when other Jobs still depend upon the Job to be deleted.

User Action: Delete all of the dependencies which refer to this Job before you delete the Job. You can list the dependencies which refer to a particular Job by selecting the “References” option from the Job Definition screen.

STILLJOBS, MONITOR shutdown requested with ## jobs still executing.

Explanation: A shutdown was requested, the monitor is shutting down. However, there were still batch jobs running whose completion will not be monitored.

User Action: If you want to maintain accurate execution history, wait until all batch jobs have completed before you shutdown the *JAMS* monitor.

STILLSUPS, There are still Setups which refer to this Job.

Explanation: You cannot delete a Job which still has Setup definitions which reference the Job.

User Action: Delete all references to the Setup before you delete the Setup.

SYSNOTFOU, System I.D. not found.

Explanation: The specified System is not in the *JAMS* database.

User Action: Enter a valid System I.D.

SYSONFILE, This System I.D. is already on file.

Explanation: You are in Add mode and the System I.D. which you entered is already defined.

User Action: Select a different System I.D. or use Modify mode.

Error Messages in order by Identifier

Error Message Format

TERMNOHIST, History not found when job terminated.

Explanation: A jobs history record could not be found in the *JAMS* history file when the job terminated.

User Action: Please submit an SPR.

TIMEOUTRNG, This Job may not be scheduled at this time.

Explanation: The Job or Setup which you are trying to submit is allowed to run only during a specific portion of the day. This time period is displayed on the screen when you are submitting the Job.

User Action: Enter a time which falls within the allowed range.

TMBUERR, Unable to set the process termination mailbox.

Explanation: The JAMS_REGISTRAR program could not set this jobs termination mailbox. Additional messages may follow this one which will indicate why the mailbox could not be set.

User Action: This error can be a result of not running the JAMS_STARTUP.COM command procedure. Make sure that the JAMS_REGISTRAR program is installed with the required privileges.

TRGDISABLED, Trigger !AS disabled.

Explanation: The specified Trigger has been disabled.

User Action: None.

TRGENABLED, Trigger !AS enabled.

Explanation: The specified Trigger has been enabled.

User Action: None.

TRGISDISABLED, Trigger !AS is already disabled.

Explanation: The specified Trigger was not disabled because it is already disabled.

User Action: None.

TRGISENABLED, Trigger !AS is already enabled.

Explanation: The specified Trigger was not enabled because it is already enabled.

User Action: None.

TRGISSET, Trigger !AS is already reset.

Explanation: The specified Trigger was not reset because it is already enabled.

User Action: None.

TRGNOTFOU, Trigger not found.

Explanation: The specified Trigger was not found.

User Action: Correct your entry.

Error Messages in order by Identifier

Error Message Format

TRGNOTRESET, Trigger !AS NOT reset, it is disabled.

Explanation: The specified Trigger was not reset because it is disabled.

User Action: Enable the Trigger.

TRGONFILE, This Trigger is already on file.

Explanation: You are trying to add a Trigger which is already on file.

User Action: Correct your entry or use Modify mode.

TRGRESET, Trigger !AS reset.

Explanation: The specified Trigger has been reset.

User Action: None.

TRYAGAIN, The precheck job will try again later.

Explanation: If a Precheck Job exits with this status, no notification is performed and the Precheck job will be resubmitted after the number of minutes specified in the Job definition.

User Action: None.

UNDEFKEY, Undefined function key.

Facility: SCU, Screen Control Utility

Explanation: You pressed a function key which does not have a defined action.

User Action: None.

USERINIT, JAMS_SCHCALLUSER initialized at dd-mmm-yyyy hh:mm

Explanation: This message will appear in the SCHEDULE.LOG log file when your JAMS_SCHCALLUSER routine is successfully initialized.

User Action: None.

VARNOTFOU, Variable not found.

Explanation: The specified Variable was not found.

User Action: Correct your entry.

VARONFILE, This Variable is already on file.

Explanation: You are trying to add a Variable which is already on file.

User Action: Correct your entry or use Modify mode.

WASEXECUTE, Job was executing, final status unknown.

Explanation: This message code is used on job history records when the final disposition of a Job is unknown.

User Action: You will see this message if JAMS was started when the job began execution but not when the job completed.

Error Messages in order by Identifier

Error Message Format

WASPENDING, Job was pending, final status is unknown.

Explanation: This message is used on job history records when a job was successfully submitted but has not been seen since.

User Action: This may see this message if *JAMS* was not started when the job actually executed. A job will also receive this status if it did not register (by running *JAMS_REGISTRAR* in the system-wide login command procedure) and the *JAMS_SCHEDULE* process was not running when the job ran.

B

Error Messages in order by Text

This appendix provides a description of the error messages which can be displayed or returned by the *JAMS* software system.

B.1 Error Message Format

JAMS uses standard VMS messages to return information to the user. Standard VMS messages have the following general format:

%FACILITY-L-IDENT, TEXT

FACILITY

A VMS facility or component name. *JAMS* messages use the facility name *JAMS*. *JAMS* may also display messages which are returned by other facilities. A percent sign (%) prefixes the first message in a sequence of related messages. Subsequent messages in the sequence are prefixed with a hyphen (-).

L

An indicator of the severity of the error. The possible values and their meaning are listed in the following table:

Code	Meaning
S	Success
I	Informational
W	Warning
E	Error
F	Fatal

IDENT

An abbreviation of the message.

TEXT

The message text. The messages in the next section are alphabetized by this message text.

Note: The messages in this appendix are arranged in alphabetical order based on the text of the message. If you want to locate a message based on the message's identifier, you can look up the identifier in the index or refer to Appendix A.

Error Messages in order by Text

Error Message Format

A Job may not depend upon itself.

Explanation: A Job may not depend upon its own completion. If a Job did depend upon its own completion and the Job ever failed, a subsequent run would remain in a pending state until manually released.

User Action: Dependencies must refer to a Job other than the one which the dependency is being defined for.

Ambiguous period specified.

Explanation: In a text date specification, you referenced a specific period which is defined for more than one date type. This frequently happens when your company uses more than one fiscal year and you define both fiscal year date types using the same specific names for each month.

User Action: You can change your date type definitions and specify unique names for the specific periods however, since these date types are already in use, this change may affect other people and/or Job definitions.

You could also use both the specific period and date type in your date specification such as "FIRST DAY OF fiscal period_01" instead of "FIRST DAY OF period_01".

Another user has changed this record while you were working.

Explanation: When you are using the *JAMS_MASTER* sub-system to update the *JAMS* database, the data you are working with is *not* locked while it is displayed on your terminal. After you have made your changes and pressed Return, the data is re-read, and a time stamp is compared to the time stamp when you initially read the data. If these two times are different it means that someone else has modified the exact same data you were working with between the time it was displayed on your terminal and the time you pressed Return.

This approach to record locking reduces locking problems but results in the possibility of this problem occurring.

User Action: When this error is detected, the data you were working with is re-read and displayed on the terminal. You may notice the changes made by the other user. You should review the data, re-enter your changes and press Return.

AST Limit reached, restarting with a larger ASTLM.

Explanation: The *JAMS_SCHEDULE* process nearly exceeded its ASTLM quota. It is restarting with a larger ASTLM quota.

User Action: None.

Clusterwide JAMS shutdown requested.

Explanation: This is an informational message. When you use the STOP MONITOR/CLUSTER command to shutdown the monitor and schedule processes on the entire cluster, this message is displayed.

User Action: None.

Error Messages in order by Text

Error Message Format

Completed, retained in queue

Explanation: This message may appear as a Jobs status in the Job Monitor. It means that the Job has completed but was retained in the queue.

User Action: None.

Completed, waiting for completion status

Explanation: This message may appear as a Jobs status in the Job Monitor. It means that the Job's process is gone but the monitor has not yet received a completion message for the JAMS_SCHEDULE process.

User Action: None.

Creating a NETWORK process on !%D.

Explanation: This message logs when a Monitor process tries to start a new JAMS_NETWORK process.

User Action: None.

Creating a SCHEDULE process on dd-mmm-yyyy.

Explanation: This message may appear in a Monitor processes log file. It is used to provide a audit trail of the creation of Schedule processes.

User Action: None.

Creating new JAMS.LOG log file in JAMS_DATA:

Explanation: This message is issued in response to the RENEW LOG_FILE command.

User Action: None.

/CURRENT is invalid for this data type.

Explanation: The /CURRENT qualifier (or option bit) is valid only for Variables with a data type of DATE, TIME or DATETIME.

User Action: Correct your code or command or change the Variables data type.

Dates for this period have not been defined.

Explanation: You have entered an English language date specification which references a valid, user defined date type but, the dates for the time period you are referencing have not been defined. For example, the date specification "1ST WORKDAY OF NEXT FISCAL NOVEMBER" may be a valid date specification but, if the start of Fiscal November of next year has not been defined, this message will be returned.

User Action: Verify that your date specification is correct. If so, then you need to define the specific dates which you are referring to.

Error Messages in order by Text

Error Message Format

Date Type not found.

Explanation: You are trying to use a Date Type which is not defined.

User Action: Enter a valid Date Type or, use the Find key to list the valid Date Types.

Declared Job MIA on dd-mmm-yyyy.

Explanation: A batch job which was being monitored by *JAMS* has disappeared. When the Monitor process discovers a missing job, it uses this message to record this abnormal event in the log file.

User Action: This should never happen. If it does, please submit an SPR.

Defined characteristic xxxxxx, number nn..

Explanation: When the Schedule process is starting, it checks for a number of queue characteristics. If these characteristics are not defined, it will define them and issue this message.

User Action: None.

Discovered missing job on dd-mmm-yyyy.

Explanation: A batch job which was executing seems to have disappeared. If the monitor cannot find the job in a reasonable period of time, it will declare the job missing in action.

User Action: If the job is subsequently declared MIA, submit an SPR.

Duplicate Parameter names on one Job are not allowed.

Explanation: You tried to define a parameter which has the same name as another parameter in this job. Parameter names must be unique within a Job.

User Action: Select a different parameter name.

Duplicate Report names on one Job are not allowed.

Explanation: You tried to define a report which has the same name as another report in this job. Report names must be unique within a Job.

User Action: Select a different report name.

Error accessing monitor mailbox.

Explanation: The JAMS_REGISTRAR program encountered an error while trying to open the mailbox to the Monitor process.

User Action: Please submit an SPR.

Error encountered when redirecting a network request.

Explanation: When a JAMS_MONITOR process was trying to redirect a connection request to the node in the VAXcluster which is running the JAMS_NETWORK process, an error was encountered. The status which caused the error is also displayed.

User Action: Please submit an SPR.

Error Messages in order by Text

Error Message Format

Errors encountered while trying to submit job.

Explanation: One or more errors were encountered while submitting a Job. Subsequent messages will describe the errors which were encountered.

User Action: Review the subsequent error messages and take appropriate action.

Error while parsing the Job file.

Explanation: *JAMS* could not parse the Job's command file. This message should be followed by additional messages which will provide more detailed information about why the command file could not be parsed. Usually, this is caused by an error in the command file or in a template customization.

User Action: Review subsequent messages and correct the problem.

Executing, process is suspended

Explanation: This message may appear as a Jobs status in the Job Monitor. It means that the Job is executing but performance information could not be obtained because the process is suspended, swapped or in a resource wait state.

User Action: None.

Fatal VMS queue manager error at dd-mmm-yyyy hh:mm

Explanation: The VMS queue manager returned an error which *JAMS* regards as fatal. The *JAMS_SCHEDULE* process will shutdown so that it can restart on a node with a working queue manager.

User Action: Check and correct your VMS queue manager.

Field is full !

Facility: SCU, Screen Control Utility

Explanation: You are trying to enter more characters into a field than the field definition allows.

User Action: None.

Function is not implemented

Explanation: This message code is returned when you call an entry point in the *JAMS* sharable image which has not been implemented.

User Action: If this message is returned by a *JAMS* program, please submit an SPR. If this message is returned to user-written code, please check your code. You can call our technical support line if you cannot determine the cause of the problem.

Error Messages in order by Text

Error Message Format

Gold...

Facility: SCU, Screen Control Utility

Explanation: You pressed the Gold key (PF1) and the system is waiting for you to press another key to complete the Gold key sequence.

User Action: If you did not intend to press the Gold key, press the Gold key again to cancel the sequence.

History not found when job terminated.

Explanation: A jobs history record could not be found in the *JAMS* history file when the job terminated.

User Action: Please submit an SPR.

/INCREMENT and /DECREMENT are invalid for this data type.

Explanation: The /INCREMENT or /DECREMENT qualifier (or option bit) is not valid for the Variables data type.

User Action: Correct your code or command or change the Variables data type.

Input is required.

Facility: SCU, Screen Control Utility

Explanation: You left a field blank and the field is a required entry.

User Action: Enter a value for the field or, press the key to abort the entry.

Insufficient memory for allocation of dynamic data areas.

Explanation: *JAMS* could not allocate additional virtual memory.

User Action: A user or system quota has been exceeded, check the PGFLQUO value in the users UAF record, also check the SYSGEN parameter VIRTUALPAGECNT and make sure that there is adequate page file space.

Internal error, please submit an SPR.

Explanation: This error should never be encountered.

User Action: Please submit an SPR.

Invalid ACK, serial number !XL, status !XL, NWIP !UL

Explanation: The message logs an invalid ACK from a remote network process. The message will be resent.

User Action: If this is a common occurrence, please submit an SPR.

Invalid answer, reply YES, NO, Y or N.

Explanation: You entered an invalid answer to a confirmation prompt.

User Action: Respond with YES, NO, Y or N.

Error Messages in order by Text

Error Message Format

Invalid batch queue specified.

Explanation: The specified queue is not a valid batch queue.

User Action: Enter a valid batch queue.

Invalid boolean value, enter T, F, Y, N, 0 or 1.

Explanation: You specified a value for a boolean Variable which is invalid.

User Action: Correct your code or command or change the Variables data type.

Invalid Character.

Facility: SCU, Screen Control Utility

Explanation: You typed a character which is invalid for the current field, such as typing a letter into a numeric field.

User Action: None.

Invalid characters in text value

Explanation: A Variable with the data type of TEXT can only contain printable ASCII characters. Any characters outside the range of ASCII 32 through 127 will cause this error.

User Action:

Invalid condition code, use the FIND key.

Explanation: You entered an invalid condition.

User Action: Use the Find key to list all of the valid conditions.

Invalid date.

Explanation: The date entered is not valid.

User Action: Enter a valid date in the form MM/DD/YY.

Invalid format.

Explanation: You entered a format which is not valid for this parameters data type. Clear the field or press the Find key to list the valid formats.

User Action: Enter a valid format.

Invalid function detected during parsing of job.

Explanation: While parsing a job, JAMS found a function reference which does not exist in the System's template library.

User Action: Correct the job file or add the text module to the template library.

Error Messages in order by Text

Error Message Format

Invalid Item List.

Explanation: The item list passed to the *JAMS* routine is not valid or is missing required information.

User Action: Correct your program.

Invalid message type of x received in the registration mailbox.

Explanation: An invalid message was placed in the *JAMS* monitor mailbox.

User Action: Please submit an SPR.

Invalid Parameter name.

Explanation: Parameter names must be valid VMS symbol names. They also must not begin with the text "JAMS_".

User Action: Enter a valid parameter name.

Invalid period.

Explanation: The date period specified is not valid.

User Action: Select a valid period.

Invalid print queue specified.

Explanation: The specified queue is not a valid print queue.

User Action: Select a valid print queue.

Invalid schedule date.

Explanation: The date specified is not a valid date specification.

User Action: Enter a valid date specification.

Invalid severity code, must be S, I, W, E or F.

Explanation: The severity code must be S (Success), I (Informational), W (Warning), E (Error) or F (Fatal).

User Action: Enter a valid severity code.

JAMS shutdown requested.

Explanation: You issued the STOP MONITOR command.

User Action: None.

JAMS_SCHCALLUSER initialized at dd-mmm-yyyy hh:mm

Explanation: This message will appear in the SCHEDULE.LOG log file when your JAMS_SCHCALLUSER routine is successfully initialized.

User Action: None.

Error Messages in order by Text

Error Message Format

Job Canceled by precheck Job.

Explanation: If a Precheck Job exits with this status, the Job which caused the Precheck to run will be deleted from the batch queues.

User Action: None.

Job has already been auto-submitted after this time.

Explanation: The JAMS_AUTOSUBMIT Job did not submit this Job because the Job has already been submitted with a scheduled time which is on or after the last time scheduled date/time for this Setup. This can happen if you re-run JAMS_AUTOSUBMIT or if you change the scheduled time and/or parameter values for JAMS_AUTOSUBMIT.

User Action: You should know what has caused this overlap of scheduled times. If you are re-running JAMS_AUTOSUBMIT because of a system crash, double check the last job in the report which has this message and make sure that it was in fact submitted.

Job not found.

Explanation: The specified job was not found in the *JAMS* database.

User Action: Enter a valid Job Name.

Job or Setup not specified in Item List.

Explanation: The item list which you passed did not contain either a Job name or a Setup name.

User Action: Correct your program.

Job should be executing.

Explanation: This is the message code is used on job history records during the time that the job is executing.

User Action: If the job is actually executing, there is no action. If the job is not executing, please submit an SPR.

Job was executing, final status unknown.

Explanation: This message code is used on job history records when the final disposition of a Job is unknown.

User Action: You will see this message if *JAMS* was started when the job began execution but not when the job completed.

Job was pending, final status is unknown.

Explanation: This message is used on job history records when a job was successfully submitted but has not been seen since.

User Action: This may see this message if *JAMS* was not started when the job actually executed. A job will also receive this status if it did not register (by running JAMS_REGISTRAR in the system-wide login command procedure) and the JAMS_SCHEDULE process was not running when the job ran.

Error Messages in order by Text

Error Message Format

Job was submitted but has not started yet.

Explanation: This message code is used on job history records between the time that a job is submitted and the time that it actually starts executing.

User Action: None.

Link error to node !AS

Explanation: A DECnet link to a remote node encountered an error. The status which caused the error is also displayed.

User Action: If the status which caused the error is not explainable, please submit an SPR.

Link replied with NAK

Explanation: A remote node replied with a NAK. The message will be resent.

User Action: If this is a common occurrence, please submit an SPR.

Link to node !AS failed

Explanation: A DECnet link to a remote node failed. The status which caused the failure is also displayed.

User Action: If the status which caused the failure is not explainable, please submit an SPR.

List has been truncated, too much data found to list it all.

Explanation: JAMS list screens have a limit of 64K bytes. If this limit is exceeded, the list is truncated so some of the requested information is not displayed.

User Action: Enter more restrictive selection criteria.

Menu not found.

Explanation: The specified Menu is not in the JAMS database.

User Action: Enter a valid Menu name.

Message from the JAMS Batch Job Monitor at dd-mmm-yyyy Abnormal termination of job xxxx, final status was...

Explanation: This message is broadcast to your terminal is a batch job terminates abnormally and you are on the notification list.

User Action: This is notification that one of your jobs died, you may want to find out why.

MONITOR shutdown requested on xxxxxx.

Explanation: A shutdown was requested, the monitor is shutting down.

User Action: None.

Error Messages in order by Text

Error Message Format

MONITOR shutdown requested on xxxx with ## jobs still executing.

Explanation: A shutdown was requested, the monitor is shutting down. However, there were still batch jobs running whose completion will not be monitored.

User Action: If you want to maintain accurate execution history, wait until all batch jobs have completed before you shutdown the *JAMS* monitor.

MONITOR shutdown requested with ## jobs still executing.

Explanation: A shutdown was requested, the monitor is shutting down. However, there were still batch jobs running whose completion will not be monitored.

User Action: If you want to maintain accurate execution history, wait until all batch jobs have completed before you shutdown the *JAMS* monitor.

MONITOR shutting down on dd-mmm-yyyy.

Explanation: A monitor shutdown was requested.

User Action: None.

MONITOR starting up on dd-mmm-yyyy.

Explanation: Logs the date and time that the monitor started.

User Action: None.

Name must start with A-Z, contain A-Z, 0-9, \$ and underscore.

Explanation: This name must be a valid VMS symbol name. This means that it must begin with an alphabetic character and contain only the characters A-Z, 0-9, \$ (dollar sign), and _ (underscore). Also the last character may not be an underscore.

User Action: Enter a valid name.

NETWORK process is not running.

Explanation: You issued the STOP NETWORK command and there is not a network process running.

User Action: None.

NETWORK process is starting up.

Explanation: This message logs the start of a net JAMS_NETWORK process.

User Action: None.

NETWORK process was started but is not responding.

Explanation: The JAMS_MONITOR process is trying to start a network process but it does not seem to be working.

User Action: Please submit an SPR.

Error Messages in order by Text

Error Message Format

No available characteristic numbers.

Explanation: *JAMS* needs to use VMS queue characteristics to control the execution of batch jobs. When the *JAMS_SCHEDULE* process starts up, it checks for the *JAMS* characteristics and will define them if they are not found.

This message is issued if *JAMS* cannot define the characteristics because there are no available characteristic numbers.

User Action: Review your VMS queue characteristics and delete at least one of them.

No History records found which match the selection criteria.

Explanation: This is self explanatory.

User Action: Modify your selection criteria. Remember that the Job Name uses VMS wildcards. Entering "ABC" will list only the Job "ABC". If you want to list all Jobs which start with "ABC", then enter "ABC*".

Also remember that the date range applies to the Jobs submit date.

No Jobs found which match the selection criteria.

Explanation: This is self explanatory.

User Action: Modify your selection criteria. Remember that the Job Name uses VMS wildcards. Entering "ABC" will list only the Job "ABC". If you want to list all Jobs which start with "ABC", then enter "ABC*".

No jobs selected for display.

Explanation: There are no jobs which meet your selection criteria. The Job monitor will re-display this message every 15-20 seconds until there are jobs which should be displayed.

User Action: None.

No Menus found which match the selection criteria.

Explanation: This is self explanatory.

User Action: Modify your selection criteria.

Node Definition not found.

Explanation: The specified Node Definition was not found in the *JAMS* database.

User Action:

Node Group not found.

Explanation: The specified Node Group was not found in the *JAMS* database.

User Action: Enter a valid Node Group Name or, use the Find key.

Error Messages in order by Text

Error Message Format

Non-workday processing must be S, I, or D. (Schedule, Ignore, Defer)

Explanation: You tried to enter an invalid value into the non-workday processing field.

User Action: The valid values are S (Schedule the job even on non-workdays), I (Ignore the job on non-workdays) or D (Defer the job until the next workday).

No Setups found which match the selection criteria.

Explanation: This is self explanatory.

User Action: Modify your selection criteria.

No Triggers found which match the selection criteria.

Explanation: There are no Triggers whose name matches the wildcard expression which you entered.

User Action: Change your wildcard expression or add the Trigger.

Not valid for this Date Type, use the Find Key.

Explanation: You entered a Specific date type which is not valid for this date.

User Action: Press the Find key to list the valid Specific Date Types.

No Variables found which match the selection criteria.

Explanation: There are no Variables whose name matches the wildcard expression which you entered.

User Action: Change your wildcard expression or add the Variable.

Obsolete file structures, please refer to the Installation Guide.

Explanation: There is a mismatch between your *JAMS* software and data file structure. This may happen when you install a new version of *JAMS* and you do not upgrade all of your *JAMS* data files.

User Action: Refer to the installation instructions for information on how to upgrade your data files. Generally this is performed by the installation procedure. If the installation procedure did not upgrade your JAMS database, the command procedure JAMS_CONVERT_DATABASE in the SYS\$UPDATE directory can be used to to this.

Press any key to continue.

Facility: SCU, Screen Control Utility

Explanation: *JAMS* is displaying some information and waiting for you to confirm receipt of the information.

User Action: "Press any key" actually means almost any key. Depending on the application, there may be active function keys.

Error Messages in order by Text

Error Message Format

Press Gold/D to confirm deletion.

Explanation: You are in delete mode and you have selected an item to be deleted. The item which you have selected is displayed for your review.

User Action: If the item which is displayed is the one which you want to delete, press Gold/D to confirm the deletion, otherwise, press any other key to abort the deletion.

SCHEDULE process is not running.

Explanation: You issued the STOP SCHEDULE command and there is not a schedule process running.

User Action: None.

Specified data is not available.

Explanation: JAMS_GET_DATA cannot return the specified data because it is not available.

User Action: None.

Received a termination message for an unregistered job.

Explanation: The JAMS_SCHEDULE process received a termination message for a job which it did not know about.

User Action: Please submit an SPR.

SCHEDULE process on !AS has been signaled to shutdown.

Explanation: Reply from the STOP SCHEDULE command.

User Action: None.

SCHEDULE process shutting down on dd-mmm-yyyy.

Explanation: Logs the date and time that the Schedule process shut down.

User Action: None.

SCHEDULE process starting up on dd-mmm-yyyy.

Explanation: Logs the date and time that the Schedule process started.

User Action: None.

SCHEDULE process was started but is not responding.

Explanation: The JAMS Schedule process has been started but is not responding. This indicates a serious problem.

User Action: Check the Schedule log file for messages which will help determine the reason for the failure of the Schedule process. The name of the log file is JAMS_DATA:SCHEDULE.LOG. If you cannot determine the cause of the schedule process failures, please submit an SPR.

Error Messages in order by Text

Error Message Format

Setup not found.

Explanation: The specified Setup is not in the *JAMS* database.

User Action: Enter a valid Setup.

Sorry, FIND is not available for this field.

Facility: SCU, Screen Control Utility

Explanation: You pressed the FIND key while the cursor was in a field which does not have a Search routine associated with it.

User Action: None.

Sorry, help is not available for this field.

Facility: SCU, Screen Control Utility

Explanation: You pressed the Help key while the cursor was in a field which does not have any help text associated with it.

User Action: Press the Help key again to obtain full screen help.

Supported datatypes are TEXT, INTEGER, DATE and TIME.

Explanation: The data type for parameters must be TEXT, INTEGER, DATE or TIME.

User Action: Enter a valid data type.

System I.D. not found.

Explanation: The specified System is not in the *JAMS* database.

User Action: Enter a valid System I.D.

The Documentation Location is not specified.

Explanation: You must specify the Documentation Location in the System Definitions before you can create documentation for a System or any of the Job's in the System.

User Action: Define the Documentation Location in the System Definition.

The end time cannot be before the scheduled time.

Explanation: The end time of a Setup which resubmits itself must be after the Setups schedule time.

User Action: If you want to have a Job resubmitted during a time span which includes midnight, you must define two Setups. One for the range before midnight and one for the range after midnight.

The JAMS NETWORK process may not be running.

Explanation: There is work for the network process but it is not running.

User Action: If the network process is in fact running, please submit an SPR.

Error Messages in order by Text

Error Message Format

The JAMS monitor has already been started.

Explanation: You tried to start the *JAMS* monitor when it was already running.

User Action: Don't do this.

The JAMS SCHEDULE process may not be running.

Explanation: The *JAMS* Schedule process is not doing its job, even though the process exists. This is not a normal condition, if it persists or occurs frequently, please submit an SPR.

User Action: Make sure that the *JAMS* schedule process is indeed running. Most of the time, this message is caused because the node which is running the schedule process has suspended operations or is hung.

If the node running the Schedule process is not hung, try stopping and restarting the Schedule process with the STOP SCHEDULE and START SCHEDULE commands.

If this does not solve the problem, stop the Schedule process and check the Schedule log file for messages which will help determine the reason for the failure of the Schedule process. The name of the log file is JAMS_DATA:SCHEDULE.LOG. If you cannot determine the cause of the schedule process failures, please submit an SPR.

The JAMS SCHEDULE process seems slow.

Explanation: The *JAMS* Schedule process is not doing its job, even though the process exists. This is not a normal condition, if it persists or occurs frequently, please submit an SPR.

User Action: Make sure that the *JAMS* schedule process is indeed running. Most of the time, this message is caused because the node which is running the schedule process has a very heavy workload and the schedule process is indeed running slowly.

The length must be between 1 and 80.

Explanation: The length of a text parameter must be between one and 80 characters long.

User Action: Enter a number between 1 and 80.

The pre-check job cannot be the same as the job itself.

Explanation: A Job cannot have itself as a Precheck Job. This would cause an endless loop because a Precheck Job could also have a Precheck job.

User Action: Correct your entry.

The precheck job will try again later.

Explanation: If a Precheck Job exits with this status, no notification is performed and the Precheck job will be resubmitted after the number of minutes specified in the Job definition.

User Action: None.

Error Messages in order by Text

Error Message Format

There are no entries in this menu.

Explanation: You selected a Menu line item but, the menu built from the selected menu definition did not have any entries.

User Action: Correct the menu definition.

There are no Parameters defined for this Job or Job Setup.

Explanation: You pressed Gold/P but this Job or Setup does not have any Parameters.

User Action: None.

There are no Reports defined for this Job.

Explanation: You pressed Gold/R but this Job does not have any Reports.

User Action: None.

There are still Jobs with this System I.D., deletion not allowed.

Explanation: You cannot delete a System which still has Jobs, Setups or Menus which refer to it.

User Action: Delete all references to the System before you delete the System.

There are still other Jobs with Dependencies which refer to this Job.

Explanation: You cannot delete a Job when other Jobs still depend upon the Job to be deleted.

User Action: Delete all of the dependencies which refer to this Job before you delete the Job. You can list the dependencies which refer to a particular Job by selecting the “References” option from the Job Definition screen.

There are still Setups which refer to this Job.

Explanation: You cannot delete a Job which still has Setup definitions which reference the Job.

User Action: Delete all references to the Setup before you delete the Setup.

The Since and Depend Job may not be the same Job.

Explanation: The Since Job and Dependent Job may not be the same because one a Job has completed twice, it has always completed since the last time it ran.

User Action: Choose a different Job for either the Since or the Dependent Job.

This Date Type does not have any Specific Types.

Explanation: You pressed the Find key to list the valid Specific Date Types for the current Date Type but this Date Type does not have any valid Specific types.

User Action: You are not allowed to enter a Specific Date Type.

Error Messages in order by Text

Error Message Format

This Date Type is already on file.

Explanation: You are in Add mode and the Date Type which you entered is already defined.

User Action: Select a different Date Type or use Modify mode.

This entry must be either Y (Yes) or N (No).

Explanation: You must enter a Y or a N. In some cases, you may leave the field blank.

User Action: Enter a Y, N or blank.

This entry must be Y (Yes), N (No) or J (Jacket).

Explanation: You entered a character other than Y, N or J.

User Action: Correct your entry.

This entry must be either B (Better) or W (Worse).

Explanation: You entered a character other than B or W.

User Action: Correct your entry.

This field must be filled.

Facility: SCU, Screen Control Utility

Explanation: You entered a value into a field which did not completely fill the field and the field must be completely filled.

User Action: Correct your entry or, blank the field.

This is not a valid VMS username, not found in SYSUAF.

Explanation: You entered a VMS username which could not be found in the default system authorization file (SYSUAF). If you want to specify a username in a System definition, you must first create the username in the SYSUAF file.

User Action: Select a different username, leave the username field blank, or add the desired username to the SYSUAF file.

This Job has already completed.

Explanation: You tried to modify a job and it has already completed.

User Action: None.

This Job is already on file.

Explanation: You are in Add mode and the Job Name which you entered is already defined.

User Action: Select a different Job Name or use Modify mode.

Error Messages in order by Text

Error Message Format

This Job is already running.

Explanation: You tried to release or reschedule a job and it is already executing.

User Action: None.

This Job is not running.

Explanation: You tried to requeue a job and it is not executing.

User Action: None.

This Job may not be scheduled at this time.

Explanation: The Job or Setup which you are trying to submit is allowed to run only during a specific portion of the day. This time period is displayed on the screen when you are submitting the Job.

User Action: Enter a time which falls within the allowed range.

This Menu is already on file.

Explanation: You are in Add mode and the Menu Name which you entered is already defined.

User Action: Select a different Menu Name or use Modify mode.

This Node Definition is already on file.

Explanation:

User Action: Enter a valid Node Definition name or, use the Find key.

This Node Group is already on file.

Explanation: You are in Add mode and the specified Node Group is already on file.

User Action: Select a different Node Group name or use Modify mode.

This Pre-Check would create a circular reference.

Explanation: The Job you specified as a Pre-check Job would create a circular list of Pre-checks. A circular list of Pre-checks is one which forms a circle of Jobs. For example, JOBA has a Pre-check of JOBB which has a Pre-check of JOBC which has a Pre-check of JOBA. If you were allowed to form a circle like this, when you submitted one of the Jobs in the circle, JAMS would continuously submit the circle of Jobs.

User Action: None.

This Setup is already on file.

Explanation: You are in Add mode and the Setup Name which you entered is already defined.

User Action: Select a different Setup Name or use Modify mode.

Error Messages in order by Text

Error Message Format

This System I.D. is already on file.

Explanation: You are in Add mode and the System I.D. which you entered is already defined.

User Action: Select a different System I.D. or use Modify mode.

This Trigger already contains this Action.

Explanation: A Trigger cannot submit the same Job more than once.

User Action: You could create a duplicate Trigger or create a Job which contains *JAMS* submit commands to submit the same Job more than once.

This Trigger already contains this Event.

Explanation: A Trigger cannot reference the same Job or Variable in two events.

User Action: If you are trying to define a Trigger with OR logic, create two separate Triggers.

This Trigger is already on file.

Explanation: You are trying to add a Trigger which is already on file.

User Action: Correct your entry or use Modify mode.

This Variable is already on file.

Explanation: You are trying to add a Variable which is already on file.

User Action: Correct your entry or use Modify mode.

Too many messages stacked up.

Explanation: The messages which the JAMS_SCHEDULE process sends out to Job Monitors were stacking up. The JAMS_SCHEDULE process will restart.

User Action: Please submit an SPR.

Trigger !AS NOT reset, it is disabled.

Explanation: The specified Trigger was not reset because it is disabled.

User Action: Enable the Trigger.

Trigger !AS disabled.

Explanation: The specified Trigger has been disabled.

User Action: None.

Trigger !AS enabled.

Explanation: The specified Trigger has been enabled.

User Action: None.

Error Messages in order by Text

Error Message Format

Trigger !AS is already disabled.

Explanation: The specified Trigger was not disabled because it is already disabled.

User Action: None.

Trigger !AS is already enabled.

Explanation: The specified Trigger was not enabled because it is already enabled.

User Action: None.

Trigger !AS is already reset.

Explanation: The specified Trigger was not reset because it is already enabled.

User Action: None.

Trigger !AS reset.

Explanation: The specified Trigger has been reset.

User Action: None.

Trigger not found.

Explanation: The specified Trigger was not found.

User Action: Correct your entry.

Unable to locate record with this Type and date.

Explanation: You are trying to view or modify a date definition and a record with the specified Date Type and date cannot be found.

User Action: Verify the data entered, use the **Find** key to display a list of dates.

Unable to locate/access job file xxxxxx

Explanation: While trying to submit a job, the *JAMS* could not locate or open the command file specified in the Job definition.

User Action: Verify your Job and System definitions. The name of the file is specified in the Job definition. The device and directory are specified in the System definition. Any logical names referenced in the file specification must be defined with the /EXECUTIVE qualifier.

Unable to obtain or define a JAMS characteristic.

Explanation: A *JAMS* procedure could not obtain or define the value of one of the JAMS characteristics. *JAMS* characteristics are defined by the JAMS_SCHEDULE process when it starts up.

User Action: Use the SHOW/QUE/CHAR command to display the queue characteristics which are defined on your system. If there are no available queue characteristic numbers, *JAMS* cannot define a new characteristic. If one of the *JAMS* characteristics is not defined and there are available characteristic numbers, make sure that the VMS Queue manager is

Error Messages in order by Text

Error Message Format

running and then use the STOP SCHEDULE and START SCHEDULE commands to restart the JAMS_SCHEDULE process.

Unable to open JAMS database file xxxxxx.

Explanation: *JAMS* could not locate, or could not open, one of the files which makes up the *JAMS* database.

User Action: You will see this problem when the *JAMS* start-up file (JAMS_STARTUP.COM) has not been executed. This procedure defines *JAMS* logical names and installs images with the privileges they will need to open the *JAMS* database.

Unable to open JCQ file.

Explanation: The JOB Completion Queue file (JCQ.DAT) could not be opened.

User Action: Make sure that the *JAMS* start-up file (JAMS_STARTUP.COM) has been executed.

Unable to open NWIP file.

Explanation: The NWIP file cannot be opened.

User Action: Please submit an SPR.

Unable to read JCQ header record at dd-mmm-yyyy hh:mm.

Explanation: The JCQ file is corrupt.

User Action: Please submit an SPR. Perform the following steps to recover your JCQ.DAT file.

- 1 Stop the Monitor and Schedule processes on all nodes in the VAXcluster.
- 2 Delete all versions of the JCQ.DAT file in the directory pointed to by the logical name JAMS_DATA.
- 3 Restart the Monitor and Schedule processes.

Unable to read JCQ record ## at dd-mmm-yyyy

Explanation: A record in the JCQ file could not be read. The RMS status value should follow this error and provide additional information about why the read failed.

User Action: Please submit an SPR.

Unable to read NWIP header record at !%D.

Explanation: The NWIP file may be corrupt.

User Action: Please submit an SPR.

Unable to read NWIP record !ZL at !%D

Explanation: The NWIP file may be corrupt.

User Action: Please submit an SPR.

Error Messages in order by Text

Error Message Format

Unable to set the process termination mailbox.

Explanation: The JAMS_REGISTRAR program could not set this jobs termination mailbox. Additional messages may follow this one which will indicate why the mailbox could not be set.

User Action: This error can be a result of not running the JAMS_STARTUP.COM command procedure. Make sure that the JAMS_REGISTRAR program is installed with the required privileges.

Unable to update JCQ header record at dd-mmm-yyyy

Explanation: The header record in the JCQ file could not be updated. The RMS status value should follow this error and provide additional information about why the update failed.

User Action: Please submit an SPR.

Unable to update NWIP header record at !%D

Explanation: The NWIP file may be corrupt.

User Action: Please submit an SPR.

Unable to write JCQ record ## at dd-mmm-yyyy

Explanation: A record in the JCQ file could not be written. The RMS status value should follow this error and provide additional information about why the write failed.

User Action: Please submit an SPR.

Unable to write NWIP record !ZL at !%D

Explanation: The NWIP file may be corrupt.

User Action: Please submit an SPR.

Undefined function key.

Facility: SCU, Screen Control Utility

Explanation: You pressed a function key which does not have a defined action.

User Action: None.

Variable not found.

Explanation: The specified Variable was not found.

User Action: Correct your entry.

Waiting for dependent Jobs to complete

Explanation: This message may appear as a Jobs status in the Job Monitor. It means that the Job is pending because not all of its dependent Jobs have completed.

User Action: None.

Error Messages in order by Text

Error Message Format

You deleted a Specific Type which is still in use !!

Explanation: While modifying a Date Type definition, you deleted a Specific Type which still has Dates defined for it.

User Action: Delete all Dates which reference the Specific Type before you delete the Specific Type.

You must delete all of the dates of this type before deleting the type.

Explanation: You tried to delete a Date Type definition which still has Dates defined for it.

User Action: Delete all Dates which reference a Date Type before you delete the Date Type.

Index

A

Abnormal Termination
 notification • 1–4, 1–25, 1–39, 1–55
ABNTerm • A–2
Access Control • 1–61, 2–50, 2–76, 2–78
Access Control Entries • 1–61
ACE • 1–61
ACQUIRE RESOURCE • 2–2
Actions • 1–44
Agent node • 1–12, 2–97
Allow Entry • 1–16
ALRAUTOSUBMIT • A–2
AMBIGPERIOD • A–2
ASTLMREACHED • A–2
Automatic Submit • 1–14, 1–34

C

Callable Interface • 4–1
CANCELJOB • A–2
CANTSUBMIT • A–2
CHARDEFINED • A–3
CHARERROR • A–3
CHECKJOBS • A–3
CIRCPRECHK • A–3
Command Files • 5–1
CONFDEL • A–3
Configuration • 1–55
/CONNECTIONS • 2–62
Copies • 1–17
CREATE DATE • 2–3
CREATE DTYPE • 2–5
CREATE METHOD • 2–6
CREATE RESOURCE • 2–8
CREATE TIME • 2–10
CREATE VARIABLE • 2–12
CRENETWORK • A–3
CRESCHED • A–4
CURRENTINV • A–4
CWSHUT • A–4

D

Data classes
 of parameters • 5–7
DATANOTAVAIL • A–4
Data Type
 of Parameters • 1–15
DATEDNOTFOU • A–4
DATENOTFOU • A–4
DATENSU • A–4
DATEONFILE • A–5
Dates • 1–53, 2–3, 2–44
 output format • 1–15
Date Specifications
 testing • 2–42
Date Types • 2–45
 defining • 1–51, 2–5
 specific • 1–52
/DEBUG • 2–62, 2–97
DECalert • 4–3
Defaults
 obtaining • 2–63, 2–99
 of parameters • 1–16
Defining
 dates • 1–53
 date types • 1–51
 dependencies • 1–18
 jobs • 1–10
 menus • 1–48
 named times • 1–50
 node groups • 1–60
 nodes • 1–59
 parameters • 1–14
 reports • 1–16
 Setups • 1–30
 systems • 1–2
 triggers • 1–43
 variables • 1–46
DELETE ENTRY • 2–28
DELETE JOB • 2–29
DELETE MENU • 2–30
DELETE METHOD • 2–31
DELETE RESOURCE • 2–32
DELETE SETUP • 2–33

Index

DELETE SYSTEM • 2–34
DELETE TIME • 2–35
DELETE TRIGGER • 2–36
DELETE VARIABLE • 2–37
Dependencies
 defining • 1–18
 releasing • 2–73
 waiting for • 2–82
/DEPENDENCIES • 2–97
Directory
 for Jobs • 1–8
 for Logs • 1–7
 for Reports • 1–8
DISABLE TIME • 2–38
DISABLE TRIGGER • 2–39
Display Only • 1–16
DUPPARAM • A–5
DUPREPORT • A–5
DUPTRGACTION • A–5
DUPTRGEVENT • A–5

E

Editor
 specifying • 1–22
ENABLE TIME • 2–40
ENABLE TRIGGER • 2–41
ENDBEFORESTART • A–5
Entries
 deleting • 2–28
 holding • 2–72
 modifying • 2–72
 releasing • 2–72
EVALUATE DATE • 2–42
Events • 1–44
EXECUTING • A–5
Execution Method • 1–21
Execution Methods
 defining • 2–6
EXTRACT DATE • 2–44
EXTRACT DTYPE • 2–45
EXTRACT JOB • 2–14, 2–46
EXTRACT MENU • 2–18, 2–47
EXTRACT METHOD • 2–48
EXTRACT RESOURCE • 2–49
EXTRACT SECURITY • 2–50
EXTRACT SETUP • 2–19, 2–52

EXTRACT SYSTEM • 2–24, 2–53
EXTRACT TIME • 2–54
EXTRACT TRIGGER • 2–26, 2–55
EXTRACT USERNAME • 2–56
EXTRACT VARIABLE • 2–57

F

FAKE COMPLETION • 2–58
FIELDFULL • A–5
FILEERR • A–6
File name
 reports • 1–17
Form • 1–17
Format
 of dates • 1–15
 of integers • 1–15
 of times • 1–15

G

GET VARIABLE • 2–60
GOLD • A–6

H

Help Text
 of parameters • 1–16
Hide field • 1–16
History
 Deleting • 3–7
 Printing • 3–17
 retention • 1–56
Holding Jobs • 2–72
Holidays • 1–14, 1–34, 1–51, 1–53

I

INCDECRINV • A–6
INPUTREQ • A–6
Integers
 output format • 1–15

Interactive
 recording • 2–65
 INTERR • A–6
 INVACK • A–6
 INVANSWER • A–6
 INVBATCHQ • A–6
 INVBOOLEAN • A–7
 INVCHAR • A–7
 INVCONDITION • A–7
 INVCTX • A–7
 INVDATAID • A–7
 INVDATATYPE • A–7
 INVDATE • A–7
 INVFORMAT • A–7
 INVFUNC • A–8
 INVIL • A–8
 INVLENGTH • A–8
 INVMBXTYPE • A–8
 INVPARAM • A–8
 INVPARSE • A–8
 INVPERIOD • A–8
 INVPRINTQ • A–8
 INVRETDSC • A–8
 INVSCHDATE • A–9
 INVSEVERITY • A–9
 INVSPECIFIC • A–9
 INVSYMNAME • A–9
 INVTEXT • A–9
 INVUSERNAME • A–9
 INVWORKDAY • A–9
 ISFMEM • A–10
 Item Lists • 4–1

J

JAMS1000 • 3–8
 JAMS1001 • 3–9
 JAMS1100 • 3–10
 JAMS1101 • 3–11
 JAMS1200 • 3–13
 JAMS2000 • 3–14
 JAMS2100 • 3–16
 JAMS3000 • 3–17
 JAMS_AUTOSUBMIT • 3–2
 JAMS_AUTOSUBMIT_INTER • 3–4
 JAMS_CLEANUP • 3–6
 JAMS_COMMAND_FILENAME • 5–5
 JAMS_DESC_LN1 • 5–5
 JAMS_DESC_LN2 • 5–5
 JAMS_EVALUATE_DATE • 4–4
 JAMS_GET_DATA • 4–6
 JAMS_GET_JOB_STATUS • 4–22
 JAMS_GET_VARIABLE • 4–28
 JAMS_HISTORY_FREE • 4–23
 JAMS_HISTORY_GET • 4–24
 JAMS_HISTORY_INIT • 4–25
 JAMS_JOB_DESC_LN1 • 5–5
 JAMS_JOB_DESC_LN2 • 5–5
 JAMS_JOB_NAME • 5–5
 JAMS_LOG_DIR • 5–5
 JAMS_LOG_SPEC • 5–5
 JAMS_MAIL_ADR • 5–5
 JAMS_NAME • 5–5
 JAMS_OPER_CLASSES • 5–5
 JAMS_PRINT_DIR • 5–5
 JAMS_PURGE • 3–7
 JAMS_REPLY_USERS • 5–5
 JAMS_RON • 5–5
 JAMS_SCHCALLUSER • 4–29
 JAMS_SCHEDULE_FREE • 4–31
 JAMS_SCHEDULE_GET_DEPEND • 4–32
 JAMS_SCHEDULE_GET_JOB • 4–33
 JAMS_SCHEDULE_INIT • 4–34
 JAMS_SCHEDULE_SORT • 4–37
 JAMS_SETUP_DESC_LN1 • 5–5
 JAMS_SETUP_DESC_LN2 • 5–5
 JAMS_SETUP_NAME • 5–5
 JAMS_SET_DATA • 4–38
 JAMS_SET_JOB_STATUS • 4–40
 JAMS_SET_VARIABLE • 4–41
 JAMS_SUBMITTED_BY • 5–6
 JAMS_SUBMIT_JOB • 4–45
 JAMS_SUBMIT_MENU • 4–43
 JAMS_SUBMIT_TIME • 5–6
 JAMS_SYSTEM_DESC • 5–6
 JAMS_SYSTEM_ID • 5–6
 JAMS_TEMP_DIR • 5–6
 JCQREADERR • A–10
 JCQUPDERR • A–10
 JCQWRITERR • A–10
 /JOB • 2–98
 Job Completion
 faking • 2–58
 Job Dependencies
 defining • 1–18
 Job Directory • 1–8

Index

JOBNIIL • A-10
JOBNOTFOU • A-10
JOBONFILE • A-10
JOBQUEERROR • A-10
Jobs • 1-10, 2-14, 2-29, 2-46
 deleting • 2-28
 dependencies • 1-18
 holding • 1-12, 2-72
 JAMS1000 • 3-8
 JAMS1001 • 3-9
 JAMS1100 • 3-10
 JAMS1101 • 3-11
 JAMS2000 • 3-14
 JAMS2100 • 3-16
 JAMS3000 • 3-17
 JAMS_AUTOSUBMIT • 3-2
 JAMS_AUTOSUBMIT_INTER • 3-4
 JAMS_CLEANUP • 3-6
 JAMS_PURGE • 3-7
 menu • 1-12
 modifying • 2-72
 parsing • 1-21
 Printing • 3-10, 3-11
 releasing • 2-72
 restartable • 1-12
 source files • 1-22
 synchronizing • 1-18
JOBSINSYS • A-11

L

License
 displaying • 2-74, 2-84
LINKERROR • A-11
LINKFAILURE • A-11
Linking • 4-2
 JAMS_SCHCALLUSER • 4-30
LINKNAK • A-11
LISTTRUNC • A-11
Log Directory • 1-7
Log Files
 Deleting • 3-6
 JAMS.LOG • 2-67
 Override Name • 1-28, 1-42
 Printing • 1-7, 1-28, 1-42
 Retaining • 1-7, 1-28, 1-42
 Time stamping • 1-42
 Timestamping • 1-7, 1-28

Logical names
 for a Report • 1-17
 JAMS_EDITOR • 1-22
 secure • 1-8

M

MBXERR • A-11
MENU Command • 2-62
MENUNOTFOU • A-11
MENUONFILE • A-11
Menus • 1-48, 2-18, 2-30, 2-47
Methods • 2-31, 2-48
 defining • 2-6
 specifying • 1-21
MIAJOB • A-12
Missed Window Action • 1-13
MJALRCOMPLETED • A-12
MJALRRUNNING • A-12
MJNOJOBS • A-12
MJNOTRUNNING • A-12
MONALRSTART • A-12
Monitor Process
 event log • 2-67
 starting • 2-90
 status of • 2-86
 stopping • 2-93
MONSHTDWN • A-12
MONSHTREQ • A-12
MONSHTRUN • A-13
MONSTART • A-13
MUSTBEBW • A-13
MUSTBEYN • A-13
MUSTFILL • A-13
Must Fill field • 1-16

N

Named Times • 2-54
 defining • 1-50, 2-10
NETNOTRESP • A-13
NETNOTRUN • A-13
NETNOTRUNNING • A-13
NETSTART • A-14
Network
 dependencies • 1-19

Network Process
 event log • 2–67
 starting • 2–91
 stopping • 2–94
 NEWLOG • A–14
 NOCHARAVAIL • A–14
 Node
 agent • 1–12
 NODEGNOTFOU • A–14
 NODEGONFILE • A–14
 Node Groups • 1–60
 NODENOTFOU • A–14
 NODEONFILE • A–14
 Nodes • 1–59, 1–60
 NODESHUT • A–14
 NODOCLOCATION • A–15
 NOFIND • A–15
 NOHELP • A–15
 NOHISTORY • A–15
 NOJCQHEAD • A–15
 NOJOBS • A–15
 NOMENUENTRIES • A–16
 NOMENUS • A–16
 NONWIPHEAD • A–16
 Non-workdays • 1–14, 1–34
 defining • 1–54
 NOPARAMS • A–16
 NOREPORTS • A–16
 NOSETUPS • A–16
 NOSPECIFIC • A–16
 Notification
 Broadcast message • 1–4, 1–25, 1–39, 1–55
 by OpenVMS Mail • 1–4, 1–25, 1–39, 1–55
 Job • 1–4, 1–25, 1–40
 of Operators • 1–4, 1–25, 1–39, 1–55
 parameters • 1–4
 Severity • 1–26, 1–41
 Times • 1–5, 1–26, 1–41
 /NOTIFY • 1–12, 1–32
 NOTIMPLEMENT • A–16
 NOTRGS • A–16
 NOVARS • A–17
 NWIPREADERR • A–17
 NWIPUPDERR • A–17
 NWIPWRITERR • A–17

O

OBSFILSTR • A–17
 OPENERR • A–17
 OPENJCQ • A–17
 OPENNWIP • A–17

P

Parameters • 1–14
 data-class • 5–7
 defaults • 2–63, 2–99
 predefined • 5–4
 referencing • 5–4, 5–6
 Parsed Jobs • 5–3
 PARSEERR • A–18
 Parsing Jobs • 1–21
 PENDING • A–18
 Precheck Job • 1–22, 1–36, 1–37
 PRECHKSAME • A–18
 Preprocessing • 1–22, 1–37
 PRESSKEY • A–18
 Print File Directory • 1–8
 Print Form • 1–17
 Printing
 History • 3–17
 Jobs • 3–10, 3–11
 Parameters • 3–11
 Reports • 3–11
 Schedules • 3–14, 3–16
 Setups • 3–13
 Systems • 3–8, 3–9
 PRINT Qualifiers • 1–18
 Priority • 1–6, 1–11
 Privileges • 4–2
 /PROMPT • 2–63, 2–98

Q

Queues
 Default Batch • 1–6
 Default for LOGS • 1–7
 Default Print • 1–8
 for a Report • 1–17
 for a specific Job • 1–11

R

RCVUNREGTRM • A-18
RECCNG • A-18
RECORD • 2-65
Recovery
 automatic • 1-23, 1-38
 instructions • 1-9, 1-24, 1-39
 job • 1-23, 1-38
Recurring Jobs • 3-2, 3-4
 automatic • 1-14
 scheduled date • 1-13
 scheduled dates/time • 1-33
 scheduled time • 1-14
REDIRERROR • A-19
RELEASE RESOURCE • 2-66
Releasing Jobs • 2-72
Remote dependencies • 1-19
Repeating Jobs • 1-14
Report Directory • 1-8
Report Files
 Deleting • 3-6
Report Headers • 3-1
Reports • 1-16
 controlling printing of • 5-9
 file name • 1-17
 printing before the end of a job • 5-9
 printing more than once • 5-9
Required field • 1-15
RESET JOB STATISTICS • 2-68
RESET SETUP STATISTICS • 2-69
RESET TRIGGER • 2-70
Resources • 2-2, 2-32, 2-49, 2-66, 2-75
 defining • 2-8
 requirements • 1-7, 1-28, 1-36
 showing • 2-85
Restart/Recovery • 1-23, 1-24, 1-38, 1-39
Restartable Jobs • 1-12
Retain Option • 1-6
Retention period for a Report • 1-18
Rexec
 Execution Method • 1-21
Runaway Job Notification • 1-5, 1-26, 1-41

S

Scheduled Date • 1-13
 default • 1-13
Scheduled Date/Time • 1-33
Scheduled Time • 1-14, 1-33
 default • 1-13
Schedule Process
 event log • 2-67
 starting • 2-90, 2-92
 status of • 2-86
 stopping • 2-93, 2-95
Schedules
 event based • 1-43
 Printing • 3-14, 3-16
 time based • 1-30
Scheduling Priority • 1-6, 1-11
SCHMSGMAX • A-19
SCHNOTRESP • A-19
SCHNOTRUN • A-19
SCHNOTRUNNING • A-19
SCHSHTDWN • A-20
SCHSHTREQ • A-20
SCHSLOW • A-20
SCHSTART • A-20
Security • 1-61, 2-50, 2-76, 2-78
 of Configuration • 1-64
 of Dates • 1-64
 of Date Types • 1-65
 of Job Monitor • 1-61
 of Jobs • 1-10, 1-62
 of Menus • 1-63
 of Named Times • 1-65
 of Node Definitions • 1-65
 of Resources • 1-66
 of Setups • 1-30, 1-62
 of Systems • 1-2, 1-63
 of Usernames • 1-66
SELFDEPEND • A-20
SET RESOURCE • 2-75
/SETUP • 2-98
SETUPNOTFOU • A-20
SETUPONFILE • A-20
Setups • 1-30, 2-19, 2-33, 2-52
 holding • 1-31
 JAMS1200 • 3-13
 menu • 1-32
 Printing • 3-13
SET VARIABLE • 2-80
/SET_SYMBOLS • 2-63, 2-98
Short Job Notification • 1-5, 1-27, 1-41

Showing Entries • 2–82
 Showing Jobs • 2–82
 SHOW RESOURCE • 2–85
 SHOW TIME • 2–87
 SHOW TRIGGER • 2–88
 SHOW VARIABLE • 2–89
 SINCEDPND • A–20
 Source files for Jobs • 1–22
 Specific Date Types • 1–52
 SPECINUSE • A–21
 Stalled Job Notification • 1–5, 1–26, 1–41
 Statistics
 resetting • 2–68, 2–69
 STILLDATES • A–21
 STILLDPND • A–21
 STILLJOBS • A–21
 STILLSUPS • A–21
 /SUBMIT • 2–63, 2–98
 Submit on hold • 1–12, 1–31, 1–45
 Suppress Menu Display • 1–12, 1–32
 Suppress Printing of a Report • 1–17
 Symbols
 for parameter values • 2–63, 2–98, 2–99
 Synchronizing jobs • 1–18
 SYSNOTFOU • A–21
 SYSONFILE • A–21
 System
 for Setup's • 1–30
 Systems • 1–2, 2–24, 2–34, 2–53
 Printing • 3–8, 3–9
 security • 1–2

T

Template Library
 specifying • 1–8
 Templates • 5–1
 TERMNOHIST • A–22
 TIMEOUTRNG • A–22
 Time Range
 for Jobs • 1–13
 for Setups • 1–32
 Times • 2–35
 defining • 1–50
 enabling • 2–38, 2–40, 2–87
 output format • 1–15
 TMBUERR • A–22

TRGDISABLED • A–22
 TRGENABLED • A–22
 TRGISDISABLED • A–22
 TRGISENABLED • A–22
 TRGISSET • A–22
 TRGNOTFOU • A–22
 TRGNOTRESET • A–23
 TRGONFILE • A–23
 TRGRESET • A–23
 Triggers • 1–43, 2–26, 2–36, 2–55
 disabling • 2–39
 enabling • 2–41
 resetting • 2–70
 showing • 2–88
 TRYAGAIN • A–23

U

UNDEFKEY • A–23
 Uppercase field • 1–16
 User exits • 4–29
 USERINIT • A–23
 Username • 1–6, 1–12, 1–32, 2–56
 User written routines • 4–29
 /USE_SYMBOLS • 2–63, 2–99

V

Variables • 1–46, 2–12, 2–37, 2–57, 2–60, 2–80, 2–89
 as default values • 1–16
 VARNOTFOU • A–23
 VARONFILE • A–23

W

WASEXECUTE • A–23
 WASPENDING • A–24
 Windows NT
 Execution Method • 1–21
 Workdays
 defining • 1–54, 1–55

