Oracle® Solaris Cluster Quorum Server Reference Manual

ORACLE®

Contents

Preface

The *Oracle Solaris Cluster Quorum Server Reference Manual* provides reference information for commands, file formats, and other public interface in Oracle Solaris Cluster Quorum Server software. This book is intended for experienced system administrators with extensive knowledge of Oracle software and hardware. This book is not to be used as a planning or presales guide. The information in this book assumes knowledge of the Solaris Operating System and expertise with the volume manager software that is used with Oracle Solaris Cluster software.

Both novice users and those familiar with the Solaris Operating System can use online man pages to obtain information about their SPARC based system or x86 based system and its features.

A man page is intended to answer concisely the question "What does this command do?" The man pages in general comprise a reference manual. They are not intended to be a tutorial.

Note – Oracle Solaris Cluster software runs on two platforms, SPARC and x86. The information in this book pertains to both platforms unless otherwise specified in a special chapter, section, note, bulleted item, figure, table, or example.

Overview

The following contains a brief description of each man page section and the information it references:

- Section 1CL describes, in alphabetical order, commands that are used for the maintenance and administration of Oracle Solaris Cluster Quorum Server.

- Section 1M describes, in alphabetical order, commands that are used chiefly for system maintenance and administration purposes.

- Section 4 outlines the formats of various files. The C structure declarations for the file formats are given where applicable.

The following is a generic format for man pages. The man pages of each manual section generally follow this order, but include only needed headings. For example, if no bugs can be reported, no BUGS section is included. See the intro pages for more information and detail about each section, and man(1) for general information about man pages.

NAME This section gives the names of the commands or functions that are documented, followed by a brief description of what they do.

SYNOPSIS This section shows the syntax of commands or functions. If a command or file does not exist in the standard path, its full path name is shown. Options and arguments are alphabetized, with single-letter arguments first, and options with arguments next, unless a different argument order is required.

The following special characters are used in this section:

[ ] Brackets. The option or argument that is enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.

. . . Ellipses. Several values can be provided for the previous argument, or the previous argument can be specified multiple times, for example, "filename...".

| Separator. Only one of the arguments separated by this character can be specified at a time.

{ } Braces. The options and/or arguments enclosed within braces are interdependent. All characters within braces must be treated as a unit.

PROTOCOL This section occurs only in subsection 3R and indicates the protocol description file.

DESCRIPTION This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. DESCRIPTION does not discuss OPTIONS or cite EXAMPLES. Interactive commands, subcommands, requests, macros, and functions are described under USAGE.

IOCTL This section appears on pages in Section 7 only. Only the device class that supplies appropriate parameters to the ioctl(2) system call is called ioctl and generates its own heading. ioctl calls for a specific device are listed alphabetically (on the man page for that specific device).

	`ioctl` calls are used for a particular class of devices. All these calls have an `io` ending, such as `mtio(7I)`.
OPTIONS	This section lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the SYNOPSIS section. Possible arguments to options are discussed under the option, and where appropriate, default values are supplied.
OPERANDS	This section lists the command operands and describes how they affect the actions of the command.
OUTPUT	This section describes the output – standard output, standard error, or output files – generated by the command.
RETURN VALUES	If the man page documents functions that return values, this section lists these values and describes the conditions under which they are returned. If a function can return only constant values, such as 0 or –1, these values are listed in tagged paragraphs. Otherwise, a single paragraph describes the return values of each function. Functions that are declared void do not return values, so they are not discussed in RETURN VALUES.
ERRORS	On failure, most functions place an error code in the global variable `errno` that indicates why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than one condition can cause the same error, each condition is described in a separate paragraph under the error code.
USAGE	This section lists special rules, features, and commands that require in-depth explanations. The subsections that are listed here are used to explain built-in functionality: Commands Modifiers Variables Expressions Input Grammar
EXAMPLES	This section provides examples of usage or of how to use a command or function. Wherever possible, a complete

example, which includes command-line entry and machine response, is shown. Whenever an example is given, the prompt is shown as example%, or if the user must be superuser, example#. Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS, and USAGE sections.

ENVIRONMENT VARIABLES	This section lists any environment variables that the command or function affects, followed by a brief description of the effect.
EXIT STATUS	This section lists the values the command returns to the calling program or shell and the conditions that cause these values to be returned. Usually, zero is returned for successful completion, and values other than zero are returned for various error conditions.
FILES	This section lists all file names that are referred to by the man page, files of interest, and files created or required by commands. Each file name is followed by a descriptive summary or explanation.
ATTRIBUTES	This section lists characteristics of commands, utilities, and device drivers by defining the attribute type and its corresponding value. See attributes(5) for more information.
SEE ALSO	This section lists references to other man pages, in-house documentation, and outside publications.
DIAGNOSTICS	This section lists diagnostic messages with a brief explanation of the condition that caused the error.
WARNINGS	This section lists warnings about special conditions that could seriously affect your working conditions. WARNINGS is not a list of diagnostics.
NOTES	This section lists additional information that does not belong anywhere else on the page. NOTES covers points of special interest to the user. Critical information is never covered here.
BUGS	This section describes known bugs and, wherever possible, suggests workarounds.

REFERENCE

OSC4QS 1cl

Name clquorumserver, clqs – manage quorum servers

Synopsis /usr/cluster/bin/clquorumserver -V

/usr/cluster/bin/clquorumserver *subcommand* -?

/usr/cluster/bin/clquorumserver *subcommand* [-v] [*quorumserver*]

/usr/cluster/bin/clquorumserver clear -c *clustername* -I *clusterID* [-y] *quorumserver*

/usr/cluster/bin/clquorumserver show [+ | *quorumserver*[...]]

/usr/cluster/bin/clquorumserver start {+ | *quorumserver*[...]}

/usr/cluster/bin/clquorumserver stop [-d] {+ | *quorumserver*[...]}

Description Use the clquorumserver command for the following tasks:

- Clean up stale configuration information of one or more quorum servers
- Display the configuration of one or more quorum servers
- Start one or more quorum servers
- Stop one or more quorum servers

The clqs command is the short form of the clquorumserver command. You can use either form of the command.

The general form of this command is as follows:

clquorumserver [*subcommand*] [*options*]

You can omit *subcommand* only if *options* specifies the -?, -v, or -V options.

The quorum server must be configured as a quorum device for the cluster. For information about configuring the quorum server, see scqsd.conf(4) and scqsd(1M). For information about adding a quorum_server type of quorum device to the cluster, see clquorum(1CL).

Subcommands The following subcommands are supported:

clear
Removes outdated cluster information from the quorum server. The quorum server keeps information about the cluster which it serves as a quorum device. This information can become invalid in the following circumstances:

- When a cluster is decommissioned without first removing the cluster quorum device using the clquorum remove command
- When a quorum_server type quorum device is removed from a cluster while the quorum server host is down

Caution – If a quorum server is not yet removed from the cluster, using this subcommand to clean up a valid quorum server could compromise the cluster quorum.

You must specify the cluster name and cluster ID for a particular quorum server. See the -c and -I options for details.

Users other than superuser require solaris.cluster.admin RBAC authorization to use this subcommand. See rbac(5) for more information.

show

Displays the configuration information about the quorum server. For every cluster that configured the quorum server as a quorum device, this subcommand shows the corresponding cluster name, cluster ID, list of reservation keys, and list of registration keys.

You can use the plus sign (+) to specify more than one quorum server.

If no operand is given, or if the plus sign (+) is specified with the operand, the command prints the configuration of all running quorum servers.

Users other than superuser require solaris.cluster.read RBAC authorization to use this subcommand. See rbac(5) for more information.

start
Starts the quorum server

stop
Stops the quorum server

Options The following options are supported:

-?
--help
Prints help information.

This option can be used alone or with a subcommand.

- If you use this option alone, the list of available subcommands is printed.
- If you use this option with a subcommand, the usage options for that subcommand are printed.

When this option is used, no other processing is performed.

-c *clustername*
--clustername *clustername*
Specifies the name of the cluster that uses the quorum server as a quorum device. You can get the cluster name by running some Oracle Solaris Cluster commands on the cluster nodes, such as cluster show.

This option is required with the clear subcommand.

-d
--disable
Disables the automatic restarting of the quorum server after a reboot.

-I *clusterID*
--clusterID *clusterID*

Specifies the cluster ID. The cluster ID is an 8-digit hexadecimal number. You can get the cluster ID by running some Oracle Solaris Cluster commands on the cluster nodes, such as `cluster show`.

This option is required with the `clear` subcommand.

-V
--version
Prints the version of the command.

Do not specify this option with subcommands, operands, or other options. The subcommand, operands, or other options are ignored. The -V option only prints the version of the command. No other operations are performed.

-v
---verbose
Prints verbose information to standard output, stdout.

You can specify this option with any form of this command.

-y
--yes
This option is only used with the `clear` subcommand. It pre-answers the confirmation question that is prompted by `clear` subcommand. Without this option, the `clear` subcommand asks a question to confirm whether you want to clean up the cluster information, and you need to answer yes or no. The subcommand only continues processing if you answer yes. When use this option, the `clear` subcommand will not ask any confirmation question, and instead directly removes the cluster information for the specified quorum server.

Operands The following operand is supported:

quorumserver
Specifies an identifier for the quorum server or servers. A quorum server can be identified by either a port number or an instance name. The port number is used by the cluster nodes to communicate with the quorum server. The instance name can be specified in the quorum server configuration file, /etc/scqsd/scqsd.conf. See `scqsd.conf(4)`.

Exit Status If the command is successful for all specified operands, it returns zero (CL_NOERR). If an error occurs for an operand, the command processes the next operand in the operand list. The returned exit code always reflects the error that occurred first.

The following exit values are returned:

0 CL_NOERR
No error

The command that you issued completed successfully.

1 CL_ENOMEM
    Not enough swap space

    A cluster node ran out of swap memory or ran out of other operating system resources.

3 CL_EINVAL
    Invalid argument

    You typed the command incorrectly, or the syntax of the cluster configuration information
    that you supplied with the -i option was incorrect.

6 CL_EACCESS
    Permission denied

    The object that you specified is inaccessible. You might need superuser or RBAC access to
    issue the command. See the su(1M) and rbac(5) man pages for more information.

18 CL_EINTERNAL
    Internal error was encountered

    An internal error indicates a software defect or other defect.

35 CL_EIO
    I/O error

    A physical input/output error has occurred.

36 CL_ENOENT
    No such object

    The object that you specified cannot be found for one of the following reasons:

    ■ The object does not exist.

    ■ A directory in the path to the configuration file that you attempted to create with the -o
      option does not exist.

    ■ The configuration file that you attempted to access with the -i option contains errors.

Examples   EXAMPLE 1   Displaying the Configuration of One Quorum Server

    The following command displays the configuration information for the quorum server that
    uses port 9000.

    # **clquorumserver show 9000**

    EXAMPLE 2   Displaying the Configuration of Several Quorum Servers

    The following command displays the configuration information for the quorum servers listed
    by their instance names.

    # **clquorumserver show qs1 qs2 qs3**

EXAMPLE 3 Displaying the Configuration of All Running Quorum Servers

The following command displays the configuration information of all running quorum servers.

```
# clquorumserver show +
```

EXAMPLE 4 Starting Quorum Servers

The following command starts all the configured quorum servers.

```
# clquorumserver start +
```

The following command starts a quorum server that is listening on port 9000.

```
# clquorumserver start 9000
```

The following command starts the quorum server instance qs1.

```
# clquorumserver start qs1
```

EXAMPLE 5 Stopping a Quorum Server By Port Number

The following command stops a quorum server that is listening on port 9000.

```
# clquorumserver stop 9000
```

EXAMPLE 6 Cleaning Up Outdated Cluster Information From the Quorum Server

This example removes information about the cluster named sc-cluster from the quorum server.

Use caution when unconfiguring a quorum server in this way. Generally, you should use clquorum remove to remove the quorum server device from cluster configuration and clean up the configuration information on the quorum server in a single set. You should only need to use this command if communications were lost between the cluster and the quorum server host during the clquorum remove operation.

```
# clquorumserver clear -c sc-cluster -I 0x4308D2CF 9000
The quorum server to be unconfigured must have been removed from
the cluster. Unconfiguring a valid quorum server could compromise
the cluster quorum. Do you want to continue? (yes or no)
```

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/service/quorum-server
Interface Stability	Evolving

See Also Intro(1CL), clquorum(1CL), cluster(1CL), scqsd(1M), scqsd.conf(4).

Notes The superuser can run all forms of this command.

Any user can run this command with the following options:

- -? (help) option
- -V (version) option

To run this command with other subcommands, users other than superuser require RBAC authorizations. See the following table.

Subcommand	RBAC Authorization
clear	solaris.cluster.admin
show	solaris.cluster.read
start	solaris.cluster.admin
stop	solaris.cluster.admin

REFERENCE

OSC4QS 1m

Name scqsd – quorum server daemon

Synopsis `/usr/cluster/lib/sc/scqsd` [-i *instance*] [-p *port-number*]
[-d *quorum-directory*]

Description The `scqsd` daemon starts automatically when the node is booted. Installing the SUNWscqsr
package sets up a startup file, /etc/scqsd/scqsd.conf, which contains information for
starting a single quorum server using default values. See the scqsd.conf(4) man page for the
format of this configuration file. Installing the package also adds an entry to the
/etc/services file. This entry specifies the default port number to be used by the quorum
server. If you need multiple quorum servers on the same machine, then create one entry per
instance in the /etc/scqsd/scqsd.conf file.

You specify options such as port number and quorum directory in the configuration file. To
start multiple quorum servers, you must provide at least a unique port number and a unique
quorum directory for each instance of the quorum server.

The use of a startup file enables administrators to start and stop different instances of the
quorum server through a single script. For details about starting and stopping the daemon on
the command line, see the clquorumserver(1CL) man page.

Examples EXAMPLE 1 Starting a Quorum Server

The following command starts a quorum server listening on port 2000.

```
# clquorumserver start 2000
```

EXAMPLE 2 Stopping All Quorum Servers

The following command stops all configured quorum servers.

```
# clquorumserver stop
```

Exit Status The following exit values are returned:

0 The command completed successfully.

nonzero An error occurred.

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	ha-cluster/service/quorum-server
Stability	Deprecated

See Also Intro(1CL), clquorumserver(1CL), scqsd.conf(4)

REFERENCE

OSC4QS 4

Name scqsd.conf – startup file for the quorum server

Synopsis `/etc/scqsd/scqsd.conf`

Description The `/etc/scqsd/scqsd.conf` file contains information that allows the `clquorumserver` command to manage quorum server instances on a machine. Each line in this file has the following format:

full-path-to-quorum-server-binary `[-i` *instancename*`]` `[-p` *portname*`]` `[-d` *quorumdirectory*`]`

Lines beginning with a number sign (#) are treated as comments and ignored. Every line is executed as specified in the file.

See scqsd(1M) for information about specific options.

Examples EXAMPLE 1   Initial scqsd.conf File

The following screen shows the default contents of the configuration file.

```
#Copyright 2005 Sun Microsystems, Inc. All rights reserved.
#Use is subject to license terms.
#
#ident "@(#)scqsd.conf 1.4  05/10/10
#
#This is the startup for for Quorum Server daemons.
#Each line in the file starts up an instance of the quorum server
#daemon. The command must have the following format:
#
#full-parth-to-quorum-server-binary 9-i instancename] [-p [port] \
#  [-d quorumdirectory]
#
#The minimum requirement to start multiple quorum servers is to
#speciry a unique port and a unique quorum directory for
#each quorum server instance.
#
#To configure more instances, add commands to this file.
#Lines beginning with a # are treated as comments and ignored.
#
/usr/cluster/lib/sc/scqsd -d /var/scqsd -p 9000
```

See Also scqsd(1M), clquorumserver(1CL)

Attributes See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	`ha-cluster/service/quorum-server`
Stability	evolving

Index