

Oracle Flash Storage System

CLI Scripting Guide



FLASH STORAGE
SYSTEMS

Part Number E68347-01
Oracle FS1-2 Flash Storage System release 6.1
2015 November

Copyright © 2005, 2015, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Contents

- List of Tables 4
- Preface 5
 - Oracle Resources 5
 - Command Syntax Conventions 5
 - Intended Audience 6
 - Scripting Support 6
- Chapter 1: FSCLI Roles and the Runtime Environment..... 7
 - Role-Based Access Control 7
 - View FSCLI Administrator Privileges in PDF and HTML 8
 - View FSCLI Administrator Privileges in FSCLI 9
 - Oracle FSCLI Environment Variables Overview 10
- Chapter 2: Oracle FSCLI Script Integration..... 14
 - Script Error Handling 14
 - Security Considerations 18
 - Script Integration Examples 18
- Chapter 3: Script Writing 22
 - Script Planning 22
 - Script Example 23
 - Script Testing 26
- Appendix A: Authorized Roles..... 28
- Index..... 41

List of Tables

- Table 1: Oracle resources.....5
- Table 2: Typography to mark command syntax.....5
- Table 3: Administrator privileges by role.....7
- Table 4: FSCLI variables.....10
- Table 5: Environment variables scripting examples.....11
- Table 6: List of common exit codes.....15
- Table 7: Special considerations for -list type commands.....16
- Table 8: Special considerations for other FSCLI requests.....17
- Table 9: Credentials integration.....19
- Table 10: Local file integration.....20
- Table 11: Session key integration.....21
- Table 12: Authorized roles for commands *a* through *c*.....28
- Table 13: Authorized roles for commands *d* through *e*.....31
- Table 14: Authorized roles for commands *f* through *l*.....33
- Table 15: Authorized roles for commands *m* through *n*.....34
- Table 16: Authorized roles for commands *o* through *r*.....35
- Table 17: Authorized roles for commands *s* through *z*.....36

Preface

Oracle Resources

Important: For the latest version of this document, visit the [SAN Storage – Oracle Flash Storage Systems](http://www.oracle.com/goto/fssystems/docs) section at the Oracle Help Center (<http://www.oracle.com/goto/fssystems/docs>).

Table 1: Oracle resources

For help with...	Contact...
Support	http://www.oracle.com/support (www.oracle.com/support)
Training	https://education.oracle.com (https://education.oracle.com)
Documentation	<ul style="list-style-type: none">• SAN Storage – Oracle Flash Storage Systems: (http://www.oracle.com/goto/fssystems/docs)• From Oracle FS System Manager (GUI): Help > Documentation• From Oracle FS System HTTP access: (http://system-name-ip/documentation.php where system-name-ip is the name or the public IP address of your system)
Documentation feedback	http://www.oracle.com/goto/docfeedback (http://www.oracle.com/goto/docfeedback)
Contact Oracle	http://www.oracle.com/us/corporate/contact/index.html (http://www.oracle.com/us/corporate/contact/index.html)

Command Syntax Conventions

Table 2: Typography to mark command syntax

Typographic symbol	Meaning
[]	Square brackets. Delimits an optional command parameter or a set of optional command parameters.
{ }	Braces. Delimits a set of command parameters, one of which must be selected.
	Vertical bar. Separates mutually exclusive parameters.

Table 2: Typography to mark command syntax (continued)

Typographic symbol	Meaning
...	Ellipsis. Indicates that the immediately preceding parameter or group of parameters can be repeated.
monospace	Indicates the name of a command or the name of a command option (sometimes called a <i>flag</i> or <i>switch</i>).
<i>italic</i>	Indicates a variable for which you need to supply a value.

Command parameters that are not enclosed within square brackets ([]) are required.

Important: The above symbols (and font styling) are based on the POSIX.1-2008 specification. These symbols are used in the command syntax only to clarify how to use the command parameters. *Do not enter these symbols on the command line.*

Intended Audience

This guide is intended for administrators of one or more Oracle Flash Storage Systems.

You must have a working knowledge of the following topics:

- The security principles and the customary practices of your business area
- A scripting language, such as BASH
- The Oracle FS System Manager (GUI)
- The Oracle FS CLI commands, the subcommands, and the options

Scripting Support

The role that Oracle plays in your script development is limited. Ensure that you understand the role that Oracle plays.

Oracle is not responsible for the correct implementation of any scripting strategies in your environment

Oracle does not support customized scripting created by your organization.

To learn more about the Oracle FS System Manager (GUI), read the *Oracle Flash Storage System Administrator's Guide* at the [Oracle Help Center](http://www.oracle.com/goto/fssystems/docs) (<http://www.oracle.com/goto/fssystems/docs>) in the Oracle Flash Storage Systems library.

To learn more about the Oracle FS CLI, read the *Oracle Flash Storage System CLI Reference* at the [Oracle Help Center](http://www.oracle.com/goto/fssystems/docs) (<http://www.oracle.com/goto/fssystems/docs>) in the Oracle Flash Storage Systems library.

FSCLI Roles and the Runtime Environment

Role-Based Access Control

The system administrator assigns each account in the Oracle FS System a specific role that defines system privileges for that account.

When you log in to the Oracle FS System with a role with insufficient privileges to perform all of the commands, all of the subcommands, and all of the options in the script, the script fails.

For example, if you create a script to add accounts and if you use an administrator account that has an *Administrator 2* role to log in to the Oracle FS System, the script fails. The script fails because only administrators who have *Primary Administrator* or *Administrator 1* roles are authorized to run the `account -add` command.

The following table lists each administrator role and a high-level overview of the privileges that an administrator having the indicated role can perform.

Table 3: Administrator privileges by role

Administrator role	Privileges
Primary Administrator	Can perform all configuration, management, and monitoring tasks, including modifying all other accounts. This account cannot be deleted or disabled.
Administrator 1	Can perform all configuration, management, and monitoring tasks except for running commands in the Drive Enclosure console or running commands using the <code>enclosure_console</code> Oracle FS CLI command.
Administrator 2	Can perform all tasks with the following caveats: <ul style="list-style-type: none"> • Cannot create or manage File Servers and administrator accounts • Cannot modify software configurations or hardware configurations • Cannot shut down the system • Cannot run commands in the Drive Enclosure console or run commands using the <code>enclosure_console</code> Oracle FS CLI command

Table 3: Administrator privileges by role (continued)

Administrator role	Privileges
Monitor	Can display system information only; cannot modify the configuration. Administrators using this role can modify their own account attributes.
Oracle Support	Can perform limited customer service-only functions; cannot modify the configuration. This account cannot be deleted or disabled. Note: Only the Oracle Customer Support personnel can use this account.
Support	Can perform only those functions that are related to customer service; cannot modify the system configuration.

View FSCLI Administrator Privileges in PDF and HTML

To determine the subcommands and the options that a specific role is authorized to perform, see the note at the end of the section that describes the command and subcommand pair. This information is available in the PDF version, in the HTML version, and in the Oracle FS CLI help.

To view the PDF version or the HTML version of the *Oracle Flash Storage System CLI Reference* guide, you need access to the [Oracle Help Center](http://www.oracle.com/goto/fssystem/docs) (<http://www.oracle.com/goto/fssystem/docs>).

- 1 Navigate to the [Oracle Help Center](http://www.oracle.com/goto/fssystem/docs) (<http://www.oracle.com/goto/fssystem/docs>).
- 2 Download the PDF or the HTML version.
- 3 Locate the command and subcommand pair for which you want to view the permissions and then locate the note at the end of that section.

```

account -add
SYNOPSIS
DESCRIPTION
Creates an account.

account -add
  -name account-name
  -role {admin1 | admin2 | monitor | support}
  [-email email-address]
  [-phone phone-number]
  [{-enable | -disable}]
  [-fullName full-name]
  [-passwordDuration duration-in-days]

  [{-sessionKey | -u admin-user -oracleFS
oracle-fs-system}]
  [{-outputformat | -o} { text | xml }]
  [{-timeout timeout-in-seconds | -verify | -usage |

```



```
-example | -help}]
```

The **account -add** command creates additional accounts. To base a new account on an existing account, run the **account -list -account account-id-or-fqn -details** command to obtain information about the roles and other settings that are currently defined for the account that you want to replicate.

Note: Only administrators with primary administrator or admin1 roles are authorized to run the **account -add** command.

View FSCLI Administrator Privileges in FSCLI

To determine the subcommands and the options that a specific role is authorized to perform, see the note at the end of the section that describes the command and subcommand pair. This information is available in the PDF version, in the HTML version, and in the Oracle FS CLI help.

- 1 At the command prompt, type the name of the command and subcommand pair for which you want to view permissions, and then include the **-help** option. For example, enter the following command and subcommand pair:

```
fscli account -add -help
```

- 2 Press the **[Enter]** key.

The following example shows the DESCRIPTION section for the Oracle FS CLI help version for the **account -add-help** command.

```
$ fscli account -add -help
account -add
    Creates an account.

SYNOPSIS
account -add
    -name account-name
    -role {admin1 | admin2 | monitor | support}
    [-email email-address]
    [-phone phone-number]
    [{-enable | -disable}]
    [-fullName full-name]
    [-passwordDuration duration-in-days]

    [{-sessionKey | -u admin-user -oracleFS oracle-
fs-system}]
    [{-outputformat | -o} { text | xml }]
    [{-timeout timeout-in-seconds | -verify | -usage
| -example | -help}]

DESCRIPTION
    The "account -add" command creates additional
    accounts. To base a new account on an existing
    account, run the "account -list -account
```

```
*account-id-or-fqn* -details" command to obtain
information about the roles and other settings
that are currently defined for the account that
you want to replicate.
```

Note: Only administrators with primary administrator or admin1 roles are authorized to run the account -add command.

Oracle FSCLI Environment Variables Overview

You can simplify your scripts by leveraging environment variables.

The following table provides a simple list of the environment variables and the command line options they replace.

Table 4: FSCLI variables

Variable	Value	Option that the variable replaces
PDS_USER	The user account ID.	-u
PDS_HOST	The name of the Oracle FS System. The value of PDS_HOST can be the DNS name or the IP address of the system.	-oracleFS
PDS_TIMEOUT	Time in seconds.	-timeout
PDS_KEY_IN_STDIN	Oracle FS System session identifier.	-sessionKey

For specifics on how each command uses the environment variables, see the *Oracle Flash Storage System CLI Reference*.

The following table lists two examples. One example of scripting without environment variables and one example of scripting with environment variables.

Important: You must place your scripts in the same directory as the FSCLI binary.

Table 5: Environment variables scripting examples

Scripting without variables	Scripting with variables
<pre> #!/usr/bin/env bash # example script using fscli credentials to add clone, # call backup, and delete clone # verify command line arguments if [\$# -ne 5]; then echo "Usage: ./\${basename \$0} <ofs-user-account> <ofs-name-or- ip> <source-lun-fqn> <backup- server-fqn> <clone-lun-name>" exit 1 fi # prompt for password read -s -p "Password: " password echo "" echo "" # check overall system status command="./fscli system -list - status -u \$1 -oraclefs \$2" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" exit \$rc fi state=\$(echo "\$response" grep - w SystemStatus awk -F': ' '{print \$2}') echo "system status is '\$state'" echo "" if [\$state != "NORMAL"] then exit \$rc fi # check source lun status command="./fscli lun -list - details -lun \$3 -u \$1 -oraclefs \$2" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" exit \$rc fi state=\$(echo "\$response" grep - w Status awk -F': ' '{print \$2}') echo "source lun status is '\$state'" echo "" if [\$state != "ONLINE"] then </pre>	<pre> #!/usr/bin/env bash # example script using fscli credentials to add clone, # call backup, and delete clone # verify command line arguments if [\$# -ne 3]; then echo "Usage: ./\${basename \$0} <source-lun-fqn> <backup-server- fqn> <clone-lun-name>" exit 1 fi # prompt for password read -s -p "Password: " password echo "" echo "" # check overall system status command="./fscli system -list - status" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" exit \$rc fi state=\$(echo "\$response" grep - w SystemStatus awk -F': ' '{print \$2}') echo "system status is '\$state'" echo "" if [\$state != "NORMAL"] then exit \$rc fi # check source lun status command="./fscli lun -list - details -lun \$1" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" exit \$rc fi state=\$(echo "\$response" grep - w Status awk -F': ' '{print \$2}') echo "source lun status is '\$state'" echo "" if [\$state != "ONLINE"] then exit \$rc fi </pre>

Table 5: Environment variables scripting examples (continued)

Scripting without variables	Scripting with variables
<pre> exit \$rc fi # check backup server status command="./fscli san_host -list - details -sanhost \$4 -u \$1 - oraclefs \$2" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" fi state=\$(echo "\$response" grep - w ManagementState awk -F': ' '{print \$2}') echo "backup server status is '\$state'" echo "" if [\$state != "AVAILABLE"] then exit \$rc fi # get next available lun number command="./fscli hostmap -list - availablelunnumbers -host \$4 -u \$1 -oraclefs \$2" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" fi lunnumber=\$(echo "\$response" grep -w LunNumber head -n1 awk -F': ' '{print \$2}') echo "next available lun number is '\$lunnumber'" echo "" # add clone command="./fscli clone_lun -add - name \$5 -source \$3 -hostmap \$4 - lunnumber \$lunnumber -u \$1 - oraclefs \$2" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" </pre>	<pre> # check backup server status command="./fscli san_host -list - details -sanhost \$2" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" fi state=\$(echo "\$response" grep - w ManagementState awk -F': ' '{print \$2}') echo "backup server status is '\$state'" echo "" if [\$state != "AVAILABLE"] then exit \$rc fi # get next available lun number command="./fscli hostmap -list - availablelunnumbers -host \$2" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" fi lunnumber=\$(echo "\$response" grep -w LunNumber head -n1 awk -F': ' '{print \$2}') echo "next available lun number is '\$lunnumber'" echo "" # add clone command="./fscli clone_lun -add - name \$3 -source \$1 -hostmap \$2 - lunnumber \$lunnumber" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" fi echo "" # check clone status </pre>

Table 5: Environment variables scripting examples (continued)

Scripting without variables	Scripting with variables
<pre> exit \$rc fi echo "" # check clone status command="./fscli clone_lun -list - details -clonelun /\$5 -u \$1 - oraclefs \$2" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" exit \$rc fi state=\$(echo "\$response" grep - w Status awk -F': ' '{print \$2}') echo "clone lun status is '\$state'" echo "" if [\$state != "ONLINE"] then exit \$rc fi # call backup application echo "call backup application" echo "backup complete" echo "" # delete clone command="./fscli clone_lun - delete -clonelun /\$5 - suppresswarnings -u \$1 -oraclefs \$2" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" exit \$rc fi echo "" echo "" </pre>	<pre> command="./fscli clone_lun -list - details -clonelun /\$3" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" exit \$rc fi state=\$(echo "\$response" grep - w Status awk -F': ' '{print \$2}') echo "clone lun status is '\$state'" echo "" if [\$state != "ONLINE"] then exit \$rc fi # call backup application echo "call backup application" echo "backup complete" echo "" # delete clone command="./fscli clone_lun - delete -clonelun /\$3 - suppresswarnings" echo \$command command="echo \$password \$command" response=\$(eval \$command) rc=\$(echo \$?) if [\$rc == 0] then echo "command passed" else echo "command failed" exit \$rc fi echo "" echo "" </pre>

Oracle FSCLI Script Integration

Script Error Handling

Even if you validate each of your commands individually before adding them to your script, errors can still occur when you execute the script. If your scripting environment handles XML parsing, consider using the global XML output option (`{-outputformat | -o} xml`) to address these errors programmatically.

The Oracle FS CLI returns errors under the following conditions: command execution errors, command errors, and fatal errors. The Oracle FS CLI returns XML output for command execution errors only.

Command execution errors

The `<ErrorList>` indicates a command execution error. The following list describes some common conditions that can cause command execution errors:

- The object specified in the command does not exist.
- The subcommand in the command is incorrect.
- The option in the command is incorrect.
- The state of the object of the command or the state of the Oracle FS System prohibits running the command.

The following code sample displays the general format of an XML command execution error message:

```
<?xml version="1.0"?>
<CLIResponse>
  <ResponseHeader>
    <ClientData>FSCLI</ClientData>
  </ResponseHeader>
  <ErrorList>
    <ErrorInformation>
      <ErrorCode>error-code-name</ErrorCode>
      <ErrorMessage>customer facing error message</ErrorMessage>
    </ErrorInformation>
  </ErrorList>
  <Status>failed</Status>
</CLIResponse>
```

Note: An error message can contain multiple `<ErrorList>...</ErrorList>` sections and `<ErrorInformation>...</ErrorInformation>` sections.

Command errors

In the case where the actual Oracle FS CLI command is wrong, Oracle FS CLI does not generate XML-formatted response even if you specify XML output. The following code sample displays the general format of a command error message.

```
<?xml version="1.0"?>
Error: Unrecognizable Command: "bad-command-name"!
Issue "fscli help" for command list.
Command Failed
```

Fatal errors

A fatal error is any error that causes the program to abort. The following list summarizes some scenarios that can cause fatal errors:

- network connectivity issues (for example, a communication issue between the Oracle FS CLI client and the Pilot)
- communication between the Oracle FS CLI client and the Oracle FS System disconnects abruptly during data transmission, causing corruption in the response received
- infrastructure errors (for example, file permission issues)

Fatal errors do not return an XML-formatted response even when requested using the global XML output option (`{-outputformat | -o} xml`). The following code sample displays the general format of a fatal error message.

```
An unexpected error occurred on the OracleFS System.
Command Failed.
```

Important: Always check the exit code before attempting to parse the response.

List of Common Exit Codes

An exit code is a one-character code that the Oracle FS System returns when there is an error either in the request sent by your script or in the response sent by the Oracle FS System.

The following table summarizes the exit codes that might be returned, their descriptions, and whether they return an XML response.

Table 6: List of common exit codes

Exit code	Description	Returns an XML response?
0	Command succeeded.	Yes, if specified.
1	Unexpected or fatal errors.	No.

Table 6: List of common exit codes (continued)

Exit code	Description	Returns an XML response?
2	Command errors, for example, incorrect or misspelled options or incorrect Oracle FS System IP addresses or DNS names.	Yes-For example, if you mistype an option name, the Oracle FS System returns XML.
		No-For example, if you fail to connect to the Oracle FS System because of a bad IP address in the command line, the Oracle FS System does not return XML.
3	Command execution errors, for example, trying to modify a LUN that does not exist.	Yes.

Use the following guidelines when checking Oracle FS System exit codes.

- Exit code zero (0) Parse XML response for an <ErrorList> tag if the command is one listed in the *Special considerations for other FSCLI requests* table below.
- Exit code two (2) or three (3) Parse XML response for an <ErrorList> tag if the error is not a fatal error.
- Exit code one (1) Do not parse the error as responses are not XML-formatted for fatal errors.

Additional Considerations

For `-list` type subcommands used with other options, the command fails unless all subcommands and options execute.

The following table lists the special consideration for the `-list` type commands when used with other options.

Table 7: Special considerations for -list type commands

Command	Description
<code>-list</code> type commands (for example <code>system -list</code>)	If you specify the <code>-system</code> command using the <code>-list</code> subcommand with any other options, the command fails unless all subcommands and options execute. You can avoid this situation if, instead of using the command <code>system -list</code> or the command <code>system -list -details</code> , use the <code>system -list -status</code> .

Some FSCLI requests issue multiple commands on the Pilot. For example, when adding a LUN and mapping the LUN to a host, the first request, creating a LUN, might succeed but the second request, mapping the LUN to a host, might fail. In this example, FSCLI returns a successful exit code zero (0) because the LUN was created but also an <ErrorList> tag (in the XML-formatted output) and a message explaining that the mapping request failed. Therefore, you must check for an <ErrorList> tag even if the overall status is exit code zero (0) to ensure that the entire command succeeded.

In this scenario, you should not retry the command as it will fail, because you are trying to recreate an object that already exists. You can avoid this situation by, instead of creating a LUN with the mapping in one command, first create a LUN, and then if the create a LUN command succeeds, then map the LUN using the LUN modify command.

The following table lists the set of commands that returns a successful exit code zero (0) upon creating an object, but which also might display an error message if a subsequent request fails.

Table 8: Special considerations for other FSCLI requests

Command	Description
<code>clone_lun -add</code>	If you include mapping options in your request to create a Clone LUN, FSCLI creates the Clone LUN first and then attempts the mapping. If the creation succeeds, the FSCLI reports successful execution of the command even if the subsequent mapping attempt fails.
<code>clone_lun -copy</code>	If you include mapping options in your request to create a copy of a Clone LUN, FSCLI creates the copy first and then attempts the mapping. If the creation succeeds, the FSCLI reports successful execution of the command even if the subsequent mapping attempt fails.
<code>lun -add</code>	If you include mapping options in your request to create a LUN, FSCLI creates the LUN first and then attempts the mapping. If the creation succeeds, the FSCLI reports successful execution of the command even if the subsequent mapping attempt fails.
<code>lun -copy</code>	If you include mapping options in your request to create a LUN copy, FSCLI creates the copy first and then attempts the mapping. If the creation succeeds, the FSCLI reports successful execution of the command even if the subsequent mapping attempt fails.

Table 8: Special considerations for other FSCLI requests (continued)

Command	Description
<code>nfs_export -add</code>	If you create an <code>nfs_export</code> using any of the options to set access, permissions or squash values, then FSCLI creates the export first, followed by the creation of an NFS host entry to configure the client access. If the creation attempt succeeds, the FSCLI reports successful execution of the command, even if the subsequent attempt to configure client access fails.
<code>san_host -add</code> <code>san_host -modify</code>	If you specify the <code>-associateGroup</code> option with a request to add a SAN host or modify an existing SAN host, FSCLI creates or modifies the SAN host first, followed by the request to associate that SAN host to a group. If the creation or modification succeeds, then the FSCLI reports successful execution of the command, even if the subsequent association fails.

Security Considerations

Follow the Oracle Security guidelines and the security principles as well as the customary practices of your business area.

The following security considerations for passwords and session keys must be taken into account when developing Oracle FS CLI scripts.

Passwords

Oracle security requirements prohibit passing passwords on the command line.

Oracle FS CLI first validates the command before prompting for the password to ensure the command, the subcommand, and any options are correctly specified.

Note: Use the `-verify` option to ensure the validity of the command syntax before you add it to your script.

Session Keys

Oracle security requirements prohibit passing session keys on the command line.

Oracle FS CLI prompts for the session key before validating and sending the command.

Note: Use the `-verify` option to ensure the validity of the command syntax before you add it to your script.

Script Integration Examples

The Oracle FS CLI provides three methods for integrating commands in scripts.

The following tables highlight the features, advantages, and disadvantages of each integration method, as well as provide an example of each of the integration methods.

Important: The example code provided here is for instructional purposes only to help you to get started with scripting for the Oracle Flash Storage System. As such, the use of this example code is unsupported.

Important: You must place your scripts in the same directory as the FSCLI binary.

Table 9: Credentials integration

Example	<pre>! /usr/bin/env bash # example script using fscli credentials # command is automatically wrapped with a login and logout # verify command line arguments if [\$# -ne 2]; then echo "Usage: ./\${basename \$0} <ofs-user- account> <ofs-name-or-ip>" exit 1 fi # prompt for password read -s -p "Password: " password # issue command and display result command="./fscli system -list -status -u \$1 - oraclefs \$2" command="echo \$password \$command" eval \$command</pre>
Features	The command is automatically wrapped with login and logout commands.
Advantages	<ul style="list-style-type: none"> • The easiest method to implement • The most extensible method • Can be used to log in and manage one or more Oracle FS System sessions concurrently • Login and logout commands are seen as atomic operations
Disadvantages	None

Table 10: Local file integration

Example	<pre> #!/usr/bin/env bash # example script using fscli local file # session key automatically managed by fscli # verify command line arguments if [\$# -ne 2]; then echo "Usage: ./\${basename \$0} <ofs-user- account> <ofs-name-or-ip>" exit 1 fi # prompt for password read -s -p "Password: " password # login command="./fscli login -u \$1 -oraclefs \$2" command="echo \$password \$command" eval \$command # issue command and display result command="./fscli system -list -status" eval \$command # logout command="./fscli logout" eval \$command </pre>
Features	The Oracle FS CLI manages the session key using a temporary file.
Advantages	Next to the Credentials Integration method, the easiest method to implement
Disadvantages	<ul style="list-style-type: none"> • Can only be used to log in to a single Oracle FS System session on a single Oracle FS System at any given time • You must take into account session error conditions when using this method. <p>Note: For the Local File integration and Session Key integration methods, you can reduce the possibility of session error conditions by using the <code>-force</code> option with each login. However, using the <code>-force</code> option might cause additional problems. For example, if a script logs in using the <code>-force</code> option for a Monitor role-based administrator account, anyone using the Oracle FS System Manager (GUI) with the same account will be logged out.</p> <ul style="list-style-type: none"> • Requires that the <code>-login</code> and <code>-logout</code> commands be added to each script • The method is not extensible

Table 11: Session key integration

Example	<pre> #!/usr/bin/env bash # example script using fscli session key # session key managed by user # verify command line arguments if [\$# -ne 2]; then echo "Usage: ./\${basename \$0} <ofs-user- account> <ofs-name-or-ip>" exit 1 fi # prompt for password read -s -p "Password: " password # login and return session key command="./fscli login -u \$1 -oraclefs \$2 - returnKey" command="echo \$password \$command" eval \$command # prompt for session key read -s -p "Sessionkey: " sessionKey # issue command and display result command="./fscli system -list -status - sessionKey" command="echo \$sessionKey \$command" eval \$command # prompt for session key read -s -p "Sessionkey: " sessionKey # logout command="./fscli logout -sessionKey" command="echo \$sessionKey \$command" eval \$command </pre>
Features	The user manages the session key.
Advantages	Can be used to log in and manage one or more Oracle FS System sessions concurrently
Disadvantages	<ul style="list-style-type: none"> • The hardest method to implement • The script must manage the session key • Requires that the script be able to handle session error conditions <p>Note: For the Local File integration and Session Key integration methods, you can reduce the possibility of session error conditions by using the <code>-force</code> option with each login. However, using the <code>-force</code> option might cause additional problems. For example, if a script logs in using the <code>-force</code> option for a Monitor role-based administrator account, anyone using the Oracle FS System Manager (GUI) with the same account will be logged out.</p>

Script Writing

Script Planning

Ensure that you take the time to plan your Oracle FS CLI script implementation before you start to write scripts.

Some commands require that you gather information before you can successfully implement the commands in your script. For example, when mapping hosts for a LUN, you need to know the fully qualified name (FQN) for the LUN.

Use the following considerations when designing your scripts.

Note: The following considerations are just guidelines. Add and subtract from the lists below as needed.

Design considerations

- Do you require cross-platform (for example Windows and UNIX) scripts?
- Do you know your corporate coding standards in regards to creating Oracle FS CLI scripts?
- Do you require structured output? If your scripting environment handles XML parsing, consider using the global XML output option (`{-outputformat | -o} xml`).
- Will you perform all management tasks from Oracle FS CLI or will you use Oracle FS CLI scripts in conjunction with Oracle FS System Manager (GUI)?
- How will you obtain the information, typically variables, that you will use in your script? For example, variables can come from the output of Oracle FS CLI commands, or a spreadsheet.

Implementation considerations

- Have you tested the syntax of all the commands using the `-verify` option before you added the commands to your script?
- Can an administrator intervene while the scripts run?
- Did you create script-specific administrator accounts? (Strongly recommended.)

Script Example

This simple example for adding a Clone will help you to get started with scripting for the Oracle FS System.

Credentials Integration Method Example Script

The following example adds a Clone, calls a backup application, and then deletes the Clone. The example is based on the Credentials Integration method and demonstrates how a single response to a password prompt permits the subsequent FSCLI commands to run. This code example also contains return code checking and some basic response parsing.

Important: The example code provided here is for instructional purposes only to help you to get started with scripting for the Oracle Flash Storage System. As such, the use of this example code is unsupported.

Important: You must place your scripts in the same directory as the FSCLI binary.

```
#!/usr/bin/env bash
# example script using fscli credentials to add clone,
# call backup, and delete clone

# verify command line arguments
if [ $# -ne 5 ]; then
    echo "Usage: ./${(basename $0)} <ofs-user-account> <ofs-name-
or-ip> <source-lun-fqn> <backup-server-fqn> <clone-lun-name>"
    exit 1
fi

# prompt for password
read -s -p "Password: " password
echo ""
echo ""

# check overall system status
command="./fscli system -list -status -u $1 -oraclefs $2"
echo $command
command="echo $password | $command"
response=$(eval $command)
rc=$(echo $?)
if [ $rc == 0 ]
then
    echo "command passed"
else
    echo "command failed"
    exit $rc
fi
state=$(echo "$response" | grep -w SystemStatus | awk -F':
' '{print $2}')
echo "system status is '$state'"
echo ""
if [ $state != "NORMAL" ]
then
    exit $rc
fi

# check source lun status
command="./fscli lun -list -details -lun $3 -u $1 -oraclefs $2"
echo $command
command="echo $password | $command"
response=$(eval $command)
rc=$(echo $?)
```

```
    if [ $rc == 0 ]
    then
        echo "command passed"
    else
        echo "command failed"
        exit $rc
    fi
state=$(echo "$response" | grep -w Status | awk -F': ' '{print $2}')
echo "source lun status is '$state'"
echo ""
    if [ $state != "ONLINE" ]
    then
        exit $rc
    fi

# check backup server status
command="./fscli san_host -list -details -sanhost $4 -u $1 -
oraclefs $2"
echo $command
command="echo $password | $command"
response=$(eval $command)
rc=$(echo $?)
    if [ $rc == 0 ]
    then
        echo "command passed"
    else
        echo "command failed"
        exit $rc
    fi
state=$(echo "$response" | grep -w ManagementState | awk -F':
 '{print $2}')
echo "backup server status is '$state'"
echo ""
    if [ $state != "AVAILABLE" ]
    then
        exit $rc
    fi

# get next available lun number
command="./fscli hostmap -list -availablelunnumbers -host $4 -u
$1 -oraclefs $2"
echo $command
command="echo $password | $command"
response=$(eval $command)
rc=$(echo $?)
    if [ $rc == 0 ]
    then
        echo "command passed"
    else
        echo "command failed"
        exit $rc
    fi
lunnumber=$(echo "$response" | grep -w LunNumber | head -n1 |
awk -F': ' '{print $2}')
echo "next available lun number is '$lunnumber'"
echo ""

# add clone
command="./fscli clone_lun -add -name $5 -source $3 -hostmap $4 -
lunnumber $lunnumber -u $1 -oraclefs $2"
echo $command
command="echo $password | $command"
response=$(eval $command)
rc=$(echo $?)
    if [ $rc == 0 ]
    then
        echo "command passed"
```



```

        else
            echo "command failed"
            exit $rc
        fi
    echo ""

    # check clone status
    command="./fscli clone_lun -list -details -clonelun /$5 -u $1 -
oraclefs $2"
    echo $command
    command="echo $password | $command"
    response=$(eval $command)
    rc=$(echo $?)
    if [ $rc == 0 ]
    then
        echo "command passed"
    else
        echo "command failed"
        exit $rc
    fi
    state=$(echo "$response" | grep -w Status | awk -F': ' '{print
$2}')
    echo "clone lun status is '$state'"
    echo ""
    if [ $state != "ONLINE" ]
    then
        exit $rc
    fi

    # call backup application
    echo "call backup application"
    echo "backup complete"
    echo ""

    # delete clone
    command="./fscli clone_lun -delete -clonelun /$5 -
suppresswarnings -u $1 -oraclefs $2"
    echo $command
    command="echo $password | $command"
    response=$(eval $command)
    rc=$(echo $?)
    if [ $rc == 0 ]
    then
        echo "command passed"
    else
        echo "command failed"
        exit $rc
    fi
    echo ""
    echo ""

```

Credentials Integration Method Output Example

The following code example is the shell output after you have run the script.

```

$ ./d_clone_lun.sh administrator OFS_MACHINE_NAME /fooLunZ /
HOST_NAME clone_fooLunZ
Password:

./fscli system -list -status -u administrator -oraclefs
OFS_MACHINE_NAME
command passed
system status is 'NORMAL'

./fscli lun -list -details -lun /fooLunZ -u administrator -
oraclefs OFS_MACHINE_NAME
command passed

```

```

source lun status is 'ONLINE'

./fscli san_host -list -details -sanhost /HOST_NAME -u
administrator -oraclefs OFS_MACHINE_NAME
command passed
backup server status is 'AVAILABLE'

./fscli hostmap -list -availablelunnumbers -host /HOST_NAME -u
administrator -oraclefs OFS_MACHINE_NAME
command passed
next available lun number is '12'

./fscli clone_lun -add -name clone_fooLunZ -source /fooLunZ -
hostmap /HOST_NAME -lunnumber 12 -u administrator -oraclefs
OFS_MACHINE_NAME
command passed

./fscli clone_lun -list -details -clonelun /clone_fooLunZ -u
administrator -oraclefs OFS_MACHINE_NAME
command passed
clone lun status is 'ONLINE'

call backup application
backup complete

./fscli clone_lun -delete -clonelun /clone_fooLunZ -
suppresswarnings -u administrator -oraclefs OFS_MACHINE_NAME
command passed

```

Script Testing

Oracle FS CLI commands can have complex option settings. Oracle FS CLI provides a mean to test individual commands before placing them in a script.

Validate the syntax of a command before adding it your script using the global `-verify` option. You can also use the global `-example` option to display a sample `-list` reply (*xml* or *text*) of the command.

-verify Option

When included with any command, the `-verify` option inspects the syntax of the command.

The `-verify` option does not perform semantic checks of option operands or execute the command. When a command fails during an Oracle FS CLI session, the Oracle FS System generates an error message and displays that error message in the command line. If there are no errors, the Oracle FS System returns a `Valid : True` statement.

The following is a code sample of the values returned using the `-verify` option.

```

$ fscli account -list -verify

Valid : True
Valid : True

```

When Oracle FS CLI checks the syntax of a command, Oracle FS CLI may make more than one call in the background. If more than one call is made, the Oracle FS CLI returns more than one validity statement. No matter how many

validity statements Oracle FS CLI returns, as long as all of them are `Valid : True`, the syntax is valid.

-example Option

To display a sample `-list` reply (`xml` or `text`) of the command, use the `-example` option. To get the XML example, add the `-output` option and the `xml` value to the command. The default example is the textual output. The following code samples demonstrate the syntax for the option as used with a command.

To display the example output for `account -list -details` in XML, use the following command:

```
account -list -details -example -outputformat xml
```

To display the example output for `account -list` in plain text, include the `-output` option and the value `text` or do not specify any `-output` option at all:

```
fscli host_group -list -details -example -outputformat text
```

```
fscli account -list -details -example
```

Note: The `-example` option only works with the `-list` subcommand. Oracle FS CLI if the `-example` option used with any other subcommand.

Authorized Roles

To increase the likelihood that your script succeeds, ensure that the user account that you specify in the script, has the proper authorizations for each command and subcommand pair as well as any relevant options.

The following tables identify the authorized roles for each command and subcommand pair. This table is useful when determining the role that an FSCLI user or script needs when running the login command to start a session that executes FSCLI commands. All of the commands in the subsequent tables are sorted in alphabetical order. Each of the tables contains the commands that fall into the indicated alphabetical range.

- [Table 12: Authorized roles for commands a through c](#)
- [Table 13: Authorized roles for commands d through e](#)
- [Table 14: Authorized roles for commands f through l](#)
- [Table 15: Authorized roles for commands m through n](#)
- [Table 16: Authorized roles for commands o through r](#)
- [Table 17: Authorized roles for commands s through z](#)

Table 12: Authorized roles for commands a through c

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
account -add	Yes.	Yes.	No.	No.	No.	No.
account -delete	Yes.	Yes.	No.	No.	No.	No.
account -forgotPassword	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
account -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
account -modify	Yes.	Yes.	Yes.	No.	Yes.	Yes.
account -resetPassword	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
call_home -list -settings	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
call_home -list -matrix	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
call_home -modify	Yes.	Yes.	No.	No.	No.	No.

Table 12: Authorized roles for commands *a* through *c* (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
call_home -reset	Yes.	Yes.	No.	No.	Yes.	Yes.
call_home -test	Yes.	Yes.	Yes.	Yes.	No.	No.
call_home -uploadMatrix	No.	No.	No.	No.	Yes.	Yes.
certificate -delete	Yes.	Yes.	No.	No.	No.	No.
certificate -download	Yes.	Yes.	No.	No.	No.	No.
certificate -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
certificate -upload	Yes.	Yes.	No.	No.	No.	No.
cifs -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
cifs -modify	Yes.	Yes.	No.	No.	No.	No.
cifs_share -add	Yes.	Yes.	No.	No.	No.	No.
cifs_share -delete	Yes.	Yes.	No.	No.	No.	No.
cifs_share -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
cifs_share -modify	Yes.	Yes.	No.	No.	No.	No.
clone_filesystem -add	Yes.	Yes.	No.	No.	No.	No.
clone_filesystem -copy	Yes.	Yes.	No.	No.	No.	No.
clone_filesystem -delete	Yes.	Yes.	No.	No.	No.	No.
clone_filesystem -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
clone_filesystem -modify	Yes.	Yes.	No.	No.	No.	No.
clone_filesystem -restore	Yes.	Yes.	No.	No.	No.	No.
clone_lun -add	Yes.	Yes.	Yes.	No.	No.	No.
clone_lun -commit	Yes.	Yes.	Yes.	No.	No.	No.
clone_lun -copy	Yes.	Yes.	Yes.	No.	No.	No.
clone_lun -delete	Yes.	Yes.	Yes.	No.	No.	No.
clone_lun -list -bs	No.	No.	No.	No.	Yes.	Yes.
clone_lun -list (All options except clone_lun -list -bs)	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
clone_lun -modify	Yes.	Yes.	Yes.	No.	No.	No.
clone_lun -prepare	Yes.	Yes.	Yes.	No.	No.	No.
clone_lun -restore	Yes.	Yes.	Yes.	No.	No.	No.

Table 12: Authorized roles for commands *a* through *c* (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
controller -beacon	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
controller -command	Yes.	Yes.	No.	No.	Yes.	Yes.
controller -discardFBM	No.	No.	No.	No.	Yes.	Yes.
controller -forceFailure	No.	No.	No.	No.	Yes.	Yes.
controller -list -master	Yes.	Yes.	No.	No.	Yes.	Yes.
controller -list (All options except controller -list -master)	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
controller -modify	Yes.	Yes.	No.	No.	No.	No.
controller -modifyAgentHaFlags	No.	No.	No.	No.	Yes.	Yes.
controller -reenable	No.	No.	No.	No.	Yes.	Yes.
controller -rejoin	No.	No.	No.	No.	Yes.	Yes.
controller -remove	No.	No.	No.	No.	Yes.	Yes.
controller -replaceFru	Yes.	Yes.	No.	No.	Yes.	Yes.
controller -reset	No.	No.	No.	No.	Yes.	Yes.
controller -completeFru	Yes.	Yes.	No.	No.	Yes.	Yes.
controller -runDiagnostics	No.	No.	No.	No.	Yes.	Yes.
drive_group -delete	Yes.	Yes.	No.	No.	Yes.	Yes.
drive_group -forceDrivesOnline	Yes.	Yes.	No.	No.	Yes.	Yes.
drive_group -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
drive_group -modify	Yes.	Yes.	No.	No.	No.	No.
drive_group -restoreDriveGroup	Yes.	Yes.	No.	No.	No.	No.
drive_group -setPersistence	Yes.	Yes.	No.	No.	Yes.	Yes.
drive_group -verifyDataConsistency	Yes.	Yes.	No.	No.	Yes.	Yes.

Table 12: Authorized roles for commands *a* through *c* (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
drive_group -acceptMisplacedDriveGroups	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
drive_group -zeroUnallocatedSpace	Yes.	Yes.	No.	No.	No.	No.

Table 13: Authorized roles for commands *d* through *e*

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
drive_group -forceDrivesOnline	Yes.	Yes.	No.	No.	Yes.	Yes.
drive_group -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
drive_group -modify	Yes.	Yes.	No.	No.	No.	No.
drive_group -restoreDriveGroup	Yes.	Yes.	No.	No.	No.	No.
drive_group -setPersistence	Yes.	Yes.	No.	No.	Yes.	Yes.
drive_group -verifyDataConsistency	Yes.	Yes.	No.	No.	Yes.	Yes.
drive_group -acceptMisplacedDriveGroups	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
drive_group -zeroUnallocatedSpace	Yes.	Yes.	No.	No.	No.	No.
enclosure -autoEnclosureOps	Yes.	Yes.	No.	No.	Yes.	Yes.
enclosure -beacon	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
enclosure -clearDriveHistory	Yes.	Yes.	No.	No.	Yes.	Yes.
enclosure -list -dataConsistency	Yes.	Yes.	No.	No.	Yes.	Yes.
enclosure -list (All options except enclosure -list -dataConsistency)	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
enclosure -modify -remove	Yes.	Yes.	No.	No.	Yes.	Yes.

Table 13: Authorized roles for commands *d* through *e* (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
enclosure -modify (All options except enclosure -modify -remove)	Yes.	Yes.	No.	No.	No.	No.
enclosure -replaceFru	Yes.	Yes.	No.	No.	Yes.	Yes.
enclosure -completeFru	Yes.	Yes.	No.	No.	Yes.	Yes.
enclosure -setDisplayNumbers	Yes.	Yes.	No.	No.	No.	No.
enclosure -download	No.	No.	No.	No.	Yes.	Yes.
enclosure_console -close	Yes.	No.	No.	No.	Yes.	Yes.
enclosure_console -open	Yes.	No.	No.	No.	Yes.	Yes.
enclosure_console -read	Yes.	No.	No.	No.	Yes.	Yes.
enclosure_console -write	Yes.	No.	No.	No.	Yes.	Yes.
errors -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
event_log -list -internal	No.	No.	No.	No.	Yes.	Yes.
event_log -list (All options except event_log -list -internal)	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
event_log -reset	No.	No.	No.	No.	Yes.	Yes.
event_notification -add	Yes.	Yes.	No.	No.	No.	No.
event_notification -delete	Yes.	Yes.	No.	No.	No.	No.
event_notification -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
event_notification -modify	Yes.	Yes.	No.	No.	No.	No.
event_notification -test	Yes.	Yes.	No.	No.	No.	No.
drive_group -delete	Yes.	Yes.	No.	No.	Yes.	Yes.

Table 14: Authorized roles for commands *f*through */*

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
fileserv -add	Yes.	Yes.	No.	No.	No.	No.
fileserv -delete	Yes.	Yes.	No.	No.	No.	No.
fileserv -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
fileserv -modify	Yes.	Yes.	No.	No.	No.	No.
filesystem -add	Yes.	Yes.	No.	No.	No.	No.
filesystem -addTier	Yes.	Yes.	No.	No.	No.	No.
filesystem -cloneDelete	Yes.	Yes.	No.	No.	No.	No.
filesystem -copy	Yes.	Yes.	No.	No.	No.	No.
filesystem -delete	Yes.	Yes.	No.	No.	No.	No.
filesystem -list -datatiers -bs	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
filesystem -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
filesystem -modify	Yes.	Yes.	No.	No.	No.	No.
filesystem -scan	Yes.	Yes.	No.	No.	No.	No.
haltpoint -add	Yes.	No.	No.	No.	Yes.	Yes.
haltpoint -delete	Yes.	No.	No.	No.	Yes.	No.
haltpoint -list -active	Yes.	Yes.	Yes.	No.	Yes.	Yes.
haltpoint -list -all	Yes.	No.	No.	No.	Yes.	Yes.
haltpoint -resume	Yes.	No.	No.	No.	Yes.	Yes.
help	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
host_group -add	Yes.	Yes.	Yes.	No.	No.	No.
host_group -delete	Yes.	Yes.	Yes.	No.	No.	No.
host_group -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
host_group -modify	Yes.	Yes.	Yes.	No.	No.	No.
hostmap -add	Yes.	Yes.	Yes.	No.	No.	No.
hostmap -delete	Yes.	Yes.	Yes.	No.	No.	No.
hostmap -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
hostmap -modify	Yes.	Yes.	Yes.	No.	No.	No.
iscsi -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
iscsi -modify	Yes.	Yes.	No.	No.	No.	No.
iscsi -ping	Yes.	Yes.	No.	No.	Yes.	Yes.
job -add	Yes.	Yes.	Yes.	No.	No.	No.

Table 14: Authorized roles for commands *f* through *l* (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
job -delete	Yes.	Yes.	Yes.	No.	No.	No.
job -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
job -modify	Yes.	Yes.	Yes.	No.	No.	No.
login	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
logout	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
lun -add	Yes.	Yes.	Yes.	No.	No.	No.
lun -cloneDelete	Yes.	Yes.	Yes.	No.	No.	No.
lun -copy	Yes.	Yes.	Yes.	No.	No.	No.
lun -delete	Yes.	Yes.	Yes.	No.	No.	No.
lun -list -bs	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
lun -list (All options except lun -list -bs)	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
lun -maximumCapacity	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
lun -modify	Yes.	Yes.	Yes.	No.	No.	No.

Table 15: Authorized roles for commands *m* through *n*

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
nas -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
nas -modify	Yes.	Yes.	No.	No.	No.	No.
nas -serviceStatus	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
nas -antivirus	Yes.	Yes.	No.	No.	No.	No.
ndmp -add	Yes.	Yes.	Yes.	No.	No.	No.
ndmp -delete	Yes.	Yes.	Yes.	No.	No.	No.
ndmp -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
ndmp -modify	Yes.	Yes.	Yes.	No.	No.	No.
ndmp -server	Yes.	Yes.	Yes.	No.	Yes.	Yes.
nfs -download	Yes.	Yes.	Yes.	Yes.	No.	Yes.
nfs -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
nfs -modify	Yes.	Yes.	No.	No.	No.	No.
nfs -refreshNlmLocks	Yes.	Yes.	No.	No.	Yes.	Yes.

Table 15: Authorized roles for commands *m* through *n* (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
nfs -upload	Yes.	Yes.	Yes.	Yes.	No.	Yes.
nfs_export -add	Yes.	Yes.	No.	No.	No.	No.
nfs_export -delete	Yes.	Yes.	No.	No.	No.	No.
nfs_export -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
nfs_export -modify	Yes.	Yes.	No.	No.	No.	No.
nfs_host -add	Yes.	Yes.	No.	No.	No.	No.
nfs_host -delete	Yes.	Yes.	No.	No.	No.	No.
nfs_host -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
nfs_host -modify	Yes.	Yes.	No.	No.	No.	No.

Table 16: Authorized roles for commands *o* through *r*

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
pilot -beacon	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
pilot -forceFailover	No.	No.	No.	No.	Yes.	Yes.
pilot -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
pilot -sendInfoToSan	Yes.	Yes.	No.	No.	No.	No.
pilot -serverHealthCheck	No.	No.	No.	No.	Yes.	Yes.
pilot -shutdown	Yes.	Yes.	No.	No.	Yes.	Yes.
profile -add	Yes.	Yes.	No.	No.	No.	No.
profile -delete	Yes.	Yes.	No.	No.	No.	No.
profile -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
quota -add	Yes.	Yes.	Yes.	No.	No.	No.
quota -collect	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
quota -collect -clearCollected	Yes.	Yes.	Yes.	No.	Yes.	Yes.
quota -delete	Yes.	Yes.	No.	No.	No.	No.
quota -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
quota -modify	Yes.	Yes.	No.	No.	No.	No.
report -delete	Yes.	Yes.	No.	No.	No.	No.
report -download	Yes.	Yes.	No.	No.	Yes.	Yes.
report -generate	Yes.	Yes.	No.	No.	No.	No.

Table 16: Authorized roles for commands *o* through *r* (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
report -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
rest -enable	Yes.	Yes.	No.	No.	Yes.	Yes.
rest -enable	Yes.	Yes.	No.	No.	Yes.	Yes.
rest -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
rest -modify	Yes.	Yes.	No.	No.	Yes.	Yes.
route -add	Yes.	Yes.	No.	No.	No.	No.
route -delete	Yes.	Yes.	No.	No.	No.	No.
route -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
route -modify	Yes.	Yes.	No.	No.	No.	No.

Table 17: Authorized roles for commands *s* through *z*

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
san -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
san -panic	No.	No.	No.	No.	Yes.	No.
sanhost -add	Yes.	Yes.	Yes.	No.	No.	No.
sanhost -delete	Yes.	Yes.	Yes.	No.	No.	No.
sanhost -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
sanhost -modify	Yes.	Yes.	Yes.	No.	No.	No.
snapshot -add	Yes.	Yes.	Yes.	No.	No.	No.
snapshot -delete	Yes.	Yes.	Yes.	No.	No.	No.
snapshot -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
snapshot -modify	Yes.	Yes.	Yes.	No.	No.	No.
snapshot -restore	Yes.	Yes.	Yes.	No.	No.	No.
snapshot_schedule -add	Yes.	Yes.	No.	No.	No.	No.
snapshot_schedule -delete	Yes.	Yes.	No.	No.	No.	No.
snapshot_schedule -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
snapshot_schedule -modify	Yes.	Yes.	No.	No.	No.	No.
snmp_host -add	Yes.	Yes.	No.	No.	No.	No.
snmp_host -delete	Yes.	Yes.	No.	No.	No.	No.
snmp_host -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.

Table 17: Authorized roles for commands s through z (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
snmp_host -modify	Yes.	Yes.	No.	No.	No.	No.
software_update -add	Yes.	Yes.	No.	No.	Yes.	Yes.
software_update -install -suppressWarnings	No.	No.	No.	No.	Yes.	Yes.
software_update -install	Yes.	Yes.	No.	No.	Yes.	Yes.
software_update -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
software_update -validate	Yes.	Yes.	No.	No.	Yes.	Yes.
statistics -delete	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
statistics -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
storage_allocation -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
storage_domain -add	Yes.	Yes.	No.	No.	No.	No.
storage_domain -cancelExcludeDriveGroup	Yes.	Yes.	No.	No.	No.	No.
storage_domain -compact	Yes.	Yes.	No.	No.	No.	No.
storage_domain -delete	Yes.	Yes.	No.	No.	No.	No.
storage_domain -excludeDriveGroup	Yes.	Yes.	No.	No.	No.	No.
storage_domain -includeDriveGroup	Yes.	Yes.	No.	No.	No.	No.
storage_domain -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
storage_domain -modify	Yes.	Yes.	No.	No.	No.	No.
storage_domain -modifyExcludeDriveGroup	Yes.	Yes.	No.	No.	No.	No.
storage_domain -verifyAutoTier	Yes.	Yes.	No.	No.	No.	No.
storage_domain -clearLostMappings	Yes.	Yes.	Yes.	No.	No.	No.
system -clearLossOfSync	Yes.	Yes.	Yes.	No.	No.	No.
system -clearLostData	Yes.	Yes.	Yes.	No.	No.	No.
system -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.

Table 17: Authorized roles for commands s through z (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
system -modify -conservativeMode -performanceMode	Yes.	Yes.	Yes.	No.	Yes.	Yes.
system -modify -enableSsh-enableEncryption	Yes.	Yes.	No.	No.	No.	No.
system -modify -enableReloadModePages -disableReloadModePages	No.	No.	No.	No.	Yes.	Yes.
system -modify	Yes.	Yes.	No.	No.	No.	No.
system -network	Yes.	Yes.	No.	No.	No.	No.
system -reset	Yes.	Yes.	No.	No.	Yes.	Yes.
system -restart -emergencyClearEsm -emergencyPreserveEsm	No.	No.	No.	No.	Yes.	Yes.
system -restart (All options except system -restart -emergencyClearEsm -emergencyPreserveEsm)	Yes.	Yes.	No.	No.	Yes.	Yes.
system -shutdown (All network options)	No.	No.	No.	No.	Yes.	Yes.
system -shutdown (All remaining options, except network options)	Yes.	Yes.	No.	No.	Yes.	Yes.
system -useBackupPersistence	No.	No.	No.	No.	Yes.	Yes.
system_alert -delete	Yes.	Yes.	No.	No.	Yes.	Yes.
system_alert -list	Yes.	Yes.	Yes.	No.	Yes.	Yes.
system_alert -threshold	Yes.	Yes.	No.	No.	No.	No.
system_log -clearLogs	No.	No.	No.	No.	Yes.	Yes.
system_log -collect -topology	No.	No.	No.	No.	Yes.	Yes.

Table 17: Authorized roles for commands s through z (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
system_log -collect (All options except system_log -collect -topology)	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
system_log -delete	No.	No.	No.	No.	Yes.	Yes.
system_log -download	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
system_log -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
system_log -send	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
system_log -uploadClientLogs	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
task -cancelRestartableTasks	No.	No.	No.	No.	Yes.	Yes.
task -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
time -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
time -modify	Yes.	Yes.	No.	No.	No.	No.
topology -generateTraffic	Yes.	Yes.	Yes.	No.	Yes.	Yes.
topology -list -violations	Yes.	Yes.	Yes.	No.	Yes.	Yes.
topology -list (All options except topology -list -violations)	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
topology -modify	Yes.	Yes.	Yes.	No.	Yes.	Yes.
topology -download	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
ups -add	Yes.	Yes.	No.	No.	No.	No.
ups -delete	Yes.	Yes.	No.	No.	No.	No.
ups -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
ups -modify	Yes.	Yes.	No.	No.	No.	No.
version -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
vif -add	Yes.	Yes.	No.	No.	No.	No.
vif -delete	Yes.	Yes.	No.	No.	No.	No.
vif -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
vif -modify	Yes.	Yes.	No.	No.	No.	No.

Table 17: Authorized roles for commands s through z (continued)

Command	Primary Admin.	Administrator 1	Administrator 2	Monitor	Oracle Support	Support
volume_group -add	Yes.	Yes.	No.	No.	No.	No.
volume_group -delete	Yes.	Yes.	No.	No.	No.	No.
volume_group -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.
volume_group -modify	Yes.	Yes.	No.	No.	No.	No.
webcli -modify	Yes.	Yes.	No.	No.	Yes.	Yes.
webcli -list	Yes.	Yes.	Yes.	Yes.	Yes.	Yes.

Index

A

administrator privileges by role
defined 7

C

contact information 5
conventions
command syntax 5
customer support 5

D

documentation
feedback 5

E

education programs 5
environment variables
host 10
PDS_HOST 10
PDS_KEY_IN-STDIN 10
PDS_USER 10
session information 10
user ID 10
error handling
scripts 14–16
errors
command errors 15
command execution errors 14
fatal errors 15
example option
how to use for testing 26, 27
examples
script integration 18
scripts 23, 25
exit codes
checking 15
defined 15

F

feedback, documentation 5
FSCLI, *see* Oracle FS System CLI (FSCLI)
FSCLI requests
special considerations 17
FSCLI scripting
intended audience 6
Oracle support for 6

H

host
environment variables 10

L

list type commands
special considerations 16

O

online help 5
Oracle documentation 5
Oracle FS System CLI (FSCLI)
PDS_HOST 10
PDS_KEY_IN-STDIN 10
PDS_USER 10
Oracle Help Center (OHC) 5

P

PDS_HOST
environment variable 10
PDS_KEY_IN-STDIN
environment variable 10
PDS_USER
environment variables 10
POSIX.1-2008 specification 5
product support 5

R

role-based access
defined 7
roles
list of authorized roles by command 28

S

sales information 5
scripts
error handling 14–16
examples 23, 25
integration examples
credentials integration 18
local file integration 18
session key integration 18
planning 22
testing 26, 27
security considerations
defined 18
for passwords 18
for session keys 18

- sessions
 - environment variables [10](#)
 - fscli [10](#)
- special considerations
 - list type commands [16](#)
 - FSCLI requests [17](#)
- Support portal [5](#)
- syntax conventions [5](#)

T

- testing
 - example option
 - how to use [26, 27](#)
 - scripts [26, 27](#)
 - verify option
 - how to use [26, 27](#)
- training programs [5](#)

U

- user ID
 - environment variables [10](#)

V

- verify option
 - how to use for testing [26, 27](#)

W

- writing scripts
 - script planning [22](#)