Pillar Axiom CLI

Reference Guide

for pdscli

ORACLE

PILLAR AXIOM

Part Number: 4420-00030-0800
Pillar Axiom release 4.1
2011-October

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

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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 which 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 on 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. 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.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu’elles soient exemptes d’erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l’accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l’utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s’applique :

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d’applications de gestion des informations. Ce logiciel ou matériel n’est pas conçu ni n’est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d’applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l’utilisation de ce logiciel ou matériel pour ce type d’applications.

Oracle et Java sont des marques déposées d’Oracle Corporation et/ou de ses affiliés.Tout autre nom mentionné peut correspondre à des marques appartenant à d’autres propriétaires qu’Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d’Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d’Advanced Micro Devices. UNIX est une marque déposée d’The Open Group.

Ce logiciel ou matériel et la documentation qui l’accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l’accès à des contenus, produits ou services tiers, ou à leur utilisation.

Table of Contents

Chapter 1 Introduction to the pdscli

Chapter 2 Manage Pillar Axiom Systems

Chapter 3 CLI Commands

Chapter 4 Management Requests

Appendix A CLI Return Status Codes

List of Tables

CHAPTER 1

Introduction to the pdscli

The Pillar Axiom command line interface (pdscli) is securely encrypted and uses an open, XML-based API to communicate with Oracle's Pillar Axiom system over a TCP/IP network using TCP port 26004. The communication is encrypted for security of administrator accounts and system information. There is no access to user filesystems or block data over the CLI.

Using the pdscli CLI client software you can:

- Use Scripts to Run Commands (p. 21)
- Use Request Input Files to Run Commands (p. 24)
- Use the Command Prompt to Run Commands (p. 26)

Note: To perform these tasks, however, you must log in using a customer administrator account.

Important! Using the *pillar* support account, you cannot perform many of those tasks. If you log in using the support account, you can issue any view type request but you cannot:

- Create or modify any other accounts.
- Delete administrator actions.
- Create, modify, or delete any data oriented resources.

The CLI complements the browser-based graphical user interface (GUI), Pillar Axiom Storage Services Manager, which includes wizards, embedded instructional text, and context-sensitive online help to simplify configuration and management. Use the CLI to:

- Run the following requests that are available only from the CLI:
  - PerformTapeRobotDiscovery (p. 504)
  - PerformResetTapeRobotDevice on p. 483

- Perform repetitive tasks through scripts. For example:
    - Monitor the state of the Pillar Axiom system periodically and perform different actions based on the system state.
    - Collect data for reports. You can combine command sequences into scripts that collect data for different audiences or different reporting needs.

Install the CLI Client Software

Note: For a list of currently-supported platforms and recent patches, consult with your account representative, or refer to the Release Notes or the *Pillar Axiom Support and Interoperability Guide*.

Download the client software to the workstation from which you intend to manage the Pillar Axiom storage system.

1. Log in to the graphical user interface (GUI).
2. Click the Support icon in the top context pane.
3. Click the Command Line Interface link in the left navigation pane.
4. Choose one of the following options from the Actions drop-down list:
    - Download CLI for Linux x86
    - Download CLI for Linux IA64
    - Download CLI for Linux x86_64
    - Download CLI for HP-UX PA-RISC
    - Download CLI for HP-UX IA64
    - Download CLI for Solaris SPARC
    - Download CLI for Solaris 10 SPARC
    - Download CLI for Solaris 10 x86
    - Download CLI for AIX
    - Download CLI for FreeBSD x86
    - Download CLI for Windows
    - Download CLI for Max x86
    - Download CLI for Max PowerPC

Extract the pdscli Executable and Supplemental Files

After you download the CLI client, use a platform-specific extraction tool to expand the compressed file. For example:

- UNIX platforms (Linux, Solaris, HP-UX, AIX) and Mac: `tar xzf <archive-file>`
- Windows: expand with the built-in XP zip compression utility or a utility such as WinZip

In addition to the `pdscli` executable, supplemental CLI files are expanded into subdirectories under the extracted executable directory:

- The cliSampleInput subdirectories contain example request files (*.req). Use them to create data input files as you Write CLI Scripts (p. 23).
- The cliSampleOutput subdirectories contain example response files (*.txt) with standard output.

  Note: If you use the sample output files to create your own code, omit any extraneous colons. Colons should separate fields only.
- The XML subdirectories contain XML-formatted copies of the example request and response files.

Display CLI Help

Run the help (p. 70) command to display CLI information:

- `pdscli help` displays a list of supported commands (see CLI Commands on p. 64).
- `pdscli help environment` displays information about configuration variables for the CLI (see Set Configuration Variables on p. 17).
- `pdscli help list-requests` displays a list of management requests that you can use with the request (p. 75) and submit (p. 82) commands (see Management Requests on p. 90).
- `pdscli help list-errors` displays a list of system error messages.

Set Configuration Variables

You can set the configuration variables in the following ways and use the settings each time you run the CLI:

- Set Configuration Variables with Command-Line Options (p. 19)
- Set Configuration Variables with Environment Variables (p. 19)
- Set Configuration Variables in a Configuration File (p. 20)

If you set a configuration variable in more than one way:

- Environment-variable settings take precedence over configuration-file settings.
- Command-line options take precedence over both of the other configuration variable options.

Table 1, CLI configuration variables, lists the CLI configuration variables. The CLI shows a value of “<not set>” until you set explicit values.

Table 1 CLI configuration variables

Variable	Options	Meaning
PDS_CONFIG	{-C <file> | --config <file>}	Identifies a configuration file that contains the variables to use in a session. Its value should be a file name, and it may include path information. If the file name is not fully qualified, the CLI will attempt to open it relative to the current working directory. **Important!** To use a configuration file, the -C and --config options must appear as the first option following the command name. Example: `pdscli submit -C cli.cfg`
PDS_TMPDIR	{-W | --tmpdir <directory>}	Identifies the directory where the CLI should write any temporary files to use while processing requests.
PDS_HOST	{-H <host> | --host <host>}	Identifies the name or IP address assigned to the management interface of the Pillar Axiom system that is to be managed.
PDS_TIMEOUT	{-T < time> | --timeout <time>}	Identifies the maximum number of seconds that are permitted for the overall request-response transaction time between the CLI client and the Pillar Axiom system.

Table 1 CLI configuration variables (continued)

Variable	Options	Meaning
PDS_ASYNC	{-A | --async}	Identifies that the Pillar Axiom system should return asynchronous responses instead of waiting for the request to complete. When this variable is set, poll the status of long-running requests to determine when requests are complete.
PDS_USER	{-u <user> | --user <user>}	Identifies an administrator username.
PDS_PASSWORD	{-p <passwd> | --password <passwd>}	Identifies the password for the administrator user name. For security reasons, do not include the PDS_USER and PDS_PASSWORD configuration variables in environment variables. If you include them in a configuration file, ensure that the file is protected from unauthorized access.
PDS_SESSIONKEY	{-s <key> | --session <key>}	Identifies a session that is established upon a successful login using the startsession (p. 79) command. Use this session key (unique ID) in conjunction with the request (p. 75) command to avoid the overhead of establishing a new session for each individual request that is submitted with the submit (p. 82) command. Be sure to use the endsession (p. 68) command to close the session when you are finished. If endsession is not used, the session will time out after the last request or response. The time out is the session timeout value.
PDS_TRACEFILE	{L <file> | --logfile <file}}	Identifies the name of a text file into which to write debug trace results. Instead of a file name, you can specify: • {- | -1} for standard output (stdout) • -2 for standard error (stderr) If not set, both standard error and standard out are sent to your current shell session.
PDS_TRACEITEM	{-I | <item > | --traceitem <item>}	Identifies the type of CLI results to write to the debug trace file. Choose from: • {1 | cmd} for command line entries • {2 | option} for processed options • {4 | xmit} for transmitted XML data • {8 | recv} for received XML data • {16 | vals} for the parsed value list • 1 to 31 bitmasks, where 31 returns all of the above (or repeat for each item to write to the log) Enter a dash (-) as a trace-item prefix to turn off that item.

Set Configuration Variables with Command-Line Options

To set CLI configuration variables with command-line options:

1. Run the following command to display a list of currently defined configuration variables:

   ```
   pdscli help environment
   ```

2. Enter the long or short options from Table 1, CLI configuration variables, at the command line to set each configuration variable to an appropriate value. Note that:

   - The -C or --config option must appear first.
   - Values that you set in this way take precedence if the variable is already set.

   For example:

   ```
   pdscli submit -I 31 -u [user] -p [password] -H [host] MyRequest
   ```

   Performs maximum debug tracing and sends "MyRequest" where using submit means you start a session, send the request, then send an endsession

Set Configuration Variables with Environment Variables

The `pdscli` CLI utility can read the values of the environment variables to simplify the input of configuration information. To set CLI configuration variables with environment variables:

1. Run the following command to display a list of currently defined configuration variables:

   ```
   pdscli help environment
   ```

2. Follow the instructions in your operating system documentation to set an environment variable. Each operating system (and shell or script environment) has its own method to set environment variables.

Set Configuration Variables in a Configuration File

This option is useful when you Write CLI Scripts (p. 23). To create a configuration file that contains values to set the CLI configuration variables:

1 Run the startsession (p. 79) command with your login information, and save the session ID that the Pillar Axiom system returns.

2 Run the following command to create a template for a plain-text configuration file:

```
pdscli help environment > cli.cfg
```

Note: You can name the file any name; `cli.cfg` is an example name only.

3 Open the `cli.cfg` file in an ASCII text editor.

4 To define values for the variables that you want to store in the configuration file, enter the name of the configuration file as the value for the PDS_CONFIG variable.

5 Remove or comment out entries for variables that should not be set.

6 Save the `cli.cfg` file.

7 To set the CLI to use the values in the configuration file, specify `-C <file>` or `--config <file>` as the first option after a command name. For example:

```
pdscli request -C cli.cfg <management-request>
```

The following example shows how to create a configuration file and use it in the UNIX Bourne shell:

1 Run the following command to create a session, and save the value in the PDS_SESSIONKEY shell variable:

```
# PDS_SESSIONKEY=`pdscli startsession
   -u administrator -p secret -H 10.0.0.1`
```

2 Run the following command to create a shell for a plain-text configuration file:

```
# pdscli help environment > /data/cli.cfg
```

3 Edit `cli.cfg` and enter values for the variables that you want to store in the configuration file. For example:

```
# cat /data/cli.cfg
   PDS_CONFIG= /data/cli.cfg
   PDS_HOST= 192.168.1.1
   PDS_USER= administrator
   PDS_PASSWORD= secret
```

```
# echo PDS_SESSIONKEY=$PDS_SESSIONKEY >>
  /data/cli.cfg
```

4 To use the cli.cfg file, either “source” the entire file or simply create a variable that defines the path and name:

```
# . /data/cli.cfg
```

or

```
# PDS_CONFIG=/data/cli.cfg; export PDS_CONFIG
```

Run Commands

CLI scripts provide repeatable ways to configure, monitor, and manage a Pillar Axiom system. See Use Scripts to Run Commands on p. 21.

If you do not need a repeatable script interface, you can use the command line interface (CLI) to:

- Generate a template, enter values for the fields in the template, and run a command to incorporate the template values into the configuration settings (see Use Request Input Files to Run Commands on p. 24)
- Run commands one at a time from a command prompt (see Use the Command Prompt to Run Commands on p. 26)

Use Scripts to Run Commands

Scripts allow you to manage a Pillar Axiom system. Be sure to:

- Design CLI Scripts (p. 22) first.
- Write CLI Scripts (p. 23) and test script functionality by running the verify (p. 88) command on individual management requests.

Design CLI Scripts

Take time to design your CLI script implementation before you start to write scripts. Table 2, Design considerations for CLI scripts, lists a few design considerations. Add your environment-specific requirements to the list so that you have a complete view before implementation.

Table 2 Design considerations for CLI scripts

Design question	Consideration

Do you need cross-platform scripts?
□ Yes
  □ Linux
  □ Windows
  □ Solaris
□ No
	Which script languages work on the platforms that you use? Consider:
	• Perl
	• Expect
	• Python
	• TCL
	Be aware of other compatibility issues too, such as line-end characters.

Do you need to adhere to corporate coding standards in CLI scripts? □ Yes □ No	Follow corporate standards. Your standards may affect which script language you use and will affect your script implementation.
Will you perform all management tasks from the CLI or will CLI scripts coexist with GUI usage? □ CLI only □ CLI and GUI	If you use both interfaces, decide which interface is better suited to which tasks. • CLI usage may be appropriate for repetitive GetAll<Objects> and Get<Object>Details requests that are used to monitor a system. • GUI usage may be appropriate for Create<Object> requests that consist of many name/value pairs.

Table 3, Implementation considerations for CLI scriptslists a few implementation considerations. Add your environment-specific requirements to the list.

Table 3 Implementation considerations for CLI scripts

Implementation question	Consideration
Is user intervention permitted while scripts run? □ Yes □ No	Consider that: • User intervention to enter a password can be inconvenient if you run scripts during off-peak hours. • Login passwords are stored in clear text in templates, configuration files, and scripts. Clear-text passwords can be a security risk.

Table 3 Implementation considerations for CLI scripts (continued)

Implementation question	Consideration
Do you need script-specific administrator accounts? ☐ Yes ☐ No	Review the administrator roles in relation to the CLI scripts that you need. Balance convenience and security: • Do you need separate administrator accounts for each person? You can create a total of 17 accounts and assign appropriate roles to them. The same accounts can access both the CLI and GUI. • Do you need one administrator account for all script usage? If so, you can create a special “script” account and assign the Type1 role to it. Use caution when creating special scripts. It is possible for multiple administrators to run the same script and interfere with each other.
Do you need script assistance from the Professional Services team at Pillar World Wide Customer Support Center? ☐ Yes ☐ No	Contact information: support@pillardata.com USA: 877.4PILLAR (877.474.5527) and ask for Professional Services at the prompt International: +1.408.518.4400 Web: http://support.pillardata.com/

Write CLI Scripts

The CLI issues requests that execute on the Pillar Axiom system during an authenticated session. The login process establishes a managed or encapsulated session based on the approach that you use.

- *Managed-session approach*. This approach provides explicit session management of extended sessions in which you perform a sequence of management requests.
  - Use the startsession (p. 79) command to log in and establish a managed session.
  - Use the request (p. 75) command to run a sequence of Management Requests (p. 90) within the managed session.
  - Use the endsession (p. 68) command to end the session.
- *Encapsulated-session approach*. This approach provides session control for single-instance commands. The session is valid only during the life of the single command. Use the submit (p. 82) command to run any one of the Management Requests (p. 90), such as a request to GetSystemStatusDetails (p. 303).

Additional command modes are available:

- The rawsend (p. 72) command mode is useful when the Pillar Axiom system is integrated with XML-based applications. The command permits direct use of XML files as input and output and eliminates the need to reformat the XML data as name/value pairs.
- The verify (p. 88) command mode is useful when you develop and test scripts or request templates. The command validates the command input without submitting the command to the Pillar Axiom system.

During script development, look for responses similar to the following example:

```
Request Failed; Status is: Failed
Error List:
   Error (4019):
      Terminated the request
   Error (4022):
      Error reserving persisted object because it is already in use
```

If you receive such responses to requests that you perform with the request (p. 75) or submit (p. 82) commands, modify the script to wait one minute and retry the request. Set the maximum number of retries to 10. This wait-and-retry sequence typically permits the system to work around Synchronous and Asynchronous Requests (p. 47) that try to operate on the same object.

Use Request Input Files to Run Commands

Some Management Requests (p. 90) have many options that are entered as <name>=<value> pairs. You can create input files that contain sets of <name>=<value> pairs and use the input files to run the request (p. 75) command.

You can call multiple input files for each management request. This permits you to group request options in meaningful ways to use a set of input files in different contexts or to run multiple requests.

The Pillar Axiom system provides input files to use as a base from which to create customized input files for your environment (see Extract the pdscli Executable and Supplemental Files on p. 16.) Look for request files (*.req) in subdirectories under the <CLI-install-dir>\cliSampleInput directory.

The input files are commented to provide information about valid values for the <name>=<value> pairs.

- Minimum and maximum values appear as shown in these lines from the ModifyFileSystemRequest.req input file for the ModifyFileSystem (p. 330) request:

  ```
  # minVal=1, maxVal=2147483647
  Capacity.CurrentCapacity = 1
  ```

  Ranges for Option Values (p. 57) provide additional information about quantities, data types, and data lengths.

- Enumerated lists appear as shown in these lines from the CreateNFSHostEntryRequest.req input file for the CreateNFSHostEntry (p. 148) request:

  ```
  # {AllHosts, SingleHost, Subnet, NISNetgroup}
  HostEntryType = AllHosts
  ```

  Note: The SYNTAX sections of the reference pages for CLI Commands (p. 64) and Management Requests (p. 90) show the enumerated list values in the following format:

  ```
  HostEntryType={AllHosts | SingleHost | Subnet |
     NISNetGroup}
  ```

- Repeatable <name>=<value> pairs appear as shown in these lines from the GetAllEventsRequest.req input file for the GetAllEvents (p. 210) request:

  ```
  # [...between 1 and 10 elements of types]
  # {Critical, Error, Warning, Informational}
  SeverityFilter.Severity[1] = Critical
  ```

  You could add a second <name>=<value> pair:

  ```
  SeverityFilter.Severity[2] = Error
  ```

Tip: To generate additional copies of commented input files, run the template (p. 86) `-d <request-name>` command.

To use a request input file to run commands:

1. Run the following command to generate a template:

   ```
   pdscli template <request-name> > input.req
   ```

   or (to generate a template with descriptive comments):

   ```
   pdscli template -d <request-name> > input.req
   ```

   - See Table 5, Requests categorized by object type, on p. 31, or Table 6, Requests categorized by action type, on p. 39, for lists of <request-names>.
   - You can name the file any name. `input.req` is an example name. The CLI requires a .req extension for the file name.

2. Open the `input.req` file in an ASCII text editor.
3. Define values for the <name>=<value> pairs in the template file.
4. Save the `input.req` file.
5. Run the startsession (p. 79) command, and note the session ID that the Pillar Axiom system returns.

   Define the session ID as an environment variable for this session (PDS_SESSIONKEY) or include the session ID as an option for each command in the managed session.

   At any time, run the help (p. 70) command for syntax guidance.

6. Run one or more request (p. 75) commands with the `{-t | --template} input.req` option.
7. Run the endsession (p. 68) command.

Use the Command Prompt to Run Commands

To run commands at the command prompt, type the command syntax in a single line. The command-prompt syntax is:

```
pdscli <command> [options] <request-name>
   <name>=<value>...
```

You may find this option useful to run the submit (p. 82) command with requests that have no or few name/value pairs, such as requests that:

- Get all (p. 42).
- Get details (p. 43).

Spaces are not permitted between the <name>, = sign, and <value> at the command prompt. When you enter a <value> that contains embedded spaces, enclose the <value> in double quote marks (").

As an alternative to extensive typing at the command prompt, you can enter the <name>=<value> pairs for a request in one or more input files. Use Request Input Files to Run Commands (p. 24) to decrease the amount of command-prompt typing that is required.

The command-prompt syntax to call an input file is:

```
pdscli <command> [options] [<request-name>]
   @input-file-name...
```

Use of Spaces in Commands

Spaces between the <name> and = sign or between the = sign and <value> are sometimes permitted and usually ignored in an input file. However, spaces are not allowed for requests entered as a single command line without a request file. As a best practice, do not use spaces. In addition, if you are entering a variable that has spaces, be sure to include the variable in quotes. For example:

```
pdscli submit -H krusty.cs.ca -u administrator -p pillar PerformUpdate
PerformUpgradeDisruptively=0
UpdatePackage=/Staged UpdateComponentName[1]=”Pilot OS”
UpdateComponentName[2]=”Pilot Software”
UpdateComponentName[3]=”Slammer Software”
UpdateComponentName[4]=”Slammer Software AX600”
UpdateComponentName[5]=”Slammer PROM”
UpdateComponentName[6]=”Slammer PROM AX600”
UpdateComponentName[7]=”Brick Firmware”
UpdateComponentName[8]=”Brick FC Firmware”
```

Types of Commands

To set variables and run submit requests:

1. Set CLI configuration variables (p. 17) for this session.
2. Run the submit (p. 82) command and specify a management request that performs the action that you need (see Table 5, Requests categorized by object type, on p. 31 or Table 6, Requests categorized by action type, on p. 39).

   At any time, run the help (p. 70) command for syntax guidance.

Technical Support

Access to Pillar Axiom systems

You manage a Pillar Axiom system by means of the standard user interfaces:

- The Pillar Axiom Storage Services Manager (GUI)
- The command line interface (CLI)

⚠ CAUTION Access by any other means is not supported and voids the warranty for your Pillar Axiom system. Remote access (ssh, telnet, ftp, and others) may also compromise data integrity.

Contacts at Pillar Data Systems

Table 4 Contacts at Pillar Data Systems

For help with...	Contact...
Error messages, usage questions, and other support issues	US and Canada: 877-4PILLAR (877-474-5527) Europe: +800 PILLAR FS (+800 74 55 27 37) Asia Pacific: +1-408-518-4515 South Africa: +0 800 980 400 Have your system serial number ready. Email: support@pillardata.com Support Portal: support.pillardata.com
Sales and general contact Information	www.pillardata.com/company/contact/
Documentation improvements and resources	docs@pillardata.com. www.pillardata.com/techdocs/—log in with your username and password.

CHAPTER 2

Manage Pillar Axiom Systems

You can manage the Pillar Axiom system through requests that operate on Pillar Axiom *objects*, which are entities that you create, modify, delete, and display. The system returns results for each request as a single response.

Note: You must create new Pillar Axiom objects in the correct sequence so that prerequisite objects are available when you create their dependent objects.

Most of the requests cannot be issued with the support account login. If you issue the request with the support user `pillar`, you will not be able to:

- Create or modify any other accounts.
- Delete administrator actions.
- Create, modify, or delete any data oriented resources.

You should check the status of a Create request to ensure it is complete and that all dependencies are satisfied before you run a dependent request. Look for a status of “Available,” and then run the dependent request. Do not run the dependent request as soon as the system acknowledges the initiation of an asynchronous request (see Synchronous and Asynchronous Requests on p. 47).

The following list identifies some requests with dependencies that must be satisfied. Verify that the:

- CreateFileServer (p. 100) request successfully completes before you run the CreateFileSystem (p. 110), CreateRoute (p. 157), or CreateVirtualInterface (p. 164) requests.
- CreateFileSystem (p. 110) request successfully completes before you run the CreateCIFSShare (p. 95) or CreateNFSExport (p. 144) requests.
- CreateNFSExport (p. 144) request successfully completes before you run the CreateNFSHostEntry (p. 148) request.

For details about recommended request sequences that observe dependencies, see System Management Task Summary (p. 49).

You can Monitor Objects (p. 52) and Manage Objects (p. 54) in any sequence after you implement the initial configuration.

Pillar Axiom Objects and Management Requests

Table 5 lists management requests categorized by types of Pillar Axiom objects.

Important! Modifying and otherwise managing objects requires an administrator login account. The support account login is limited to the following functions:

- Monitor data and administrative objects
- Modify Pillar Axiom software and firmware object
- Perform diagnostic and recovery functions

Table 5 Requests categorized by object type

Object	Requests
Filesystems (see also Snap FSs, filesystem copies, and CloneFSs on p. 34	• CancelForceFileSystemRebalance (p. 91) • CreateFileSystem (p. 110) • DeleteFileSystem (p. 176) • DeleteVolumeAndChildren (p. 193) • ForceFileSystemRebalanceOnRestart (p. 196) • GetAllFileSystems (p. 214) • GetFileSystemDetails (p. 255) • GetFileSystemStatistics (p. 260) • GetHistoricalFileSystemStatistics (p. 263) • ModifyFileSystem (p. 330) • PerformFileSystemCheck (p. 423) (if “Filesystem corrupted” appears in a GetAllAdministratorActions (p. 201) response) • RestoreFileSystemFromCloneFS (p. 526)
Modeler	• GetStorageConfig (p. 300) • GetNASConfigDetails (p. 274) • GetSANConfigDetails (p. 284) • GetSystemStatusDetails (p. 303)

Table 5 Requests categorized by object type (continued)

Object	Requests
NAS	• GetNASConfigDetails (p. 274) • GetAllFileSystems (p. 214) • GetFileSystemDetails (p. 255) • GetStorageConfig (p. 300) • ModifyManagementConfig (p. 360)
SAN (see also Snap LUNs and LUN copies on p. 34*)*	• GetAllSANProtocolStatistics (p. 228) • GetSANConfigDetails (p. 284) • GetSANConfigDetails on p. 284 Hosts • DeleteSANHost (p. 189) • GetAllSANHosts (p. 227) • GetHostMappingConfigDetails (p. 265) • GetLUNMappingConfigDetails (p. 270) • GetSANPortDetails (p. 286) • GetSANSlammerPortDetails (p. 290) • ModifySANHost (p. 377) HBA ports: • GetHBAPortConnections (p. 262) LUNs: • GetAllAvailableLUNNumbers (p. 202)
NFS exports*)*	• CreateNFSExport (p. 144) • GetNFSExportDetails (p. 276) • GetAllNFSExports (p. 222) • GetNFSExportDetails (p. 276) NFS host entry: • CreateNFSHostEntry (p. 148) • ModifyNFSHostEntry (p. 371) • DeleteNFSHostEntry (p. 184)
CIFS local groups	• PerformAssignCIFSLocalGroups (p. 397)
CIFS shares*)*	• CreateCIFSShare (p. 95) • ModifyCIFSShare (p. 315) • DeleteCIFSShare (p. 171) • GetAllCIFSShares (p. 206) • GetCIFSShareDetails (p. 244)
SNMP*)*	• ModifySNMPConfig (p. 384) • GetSNMPConfigDetails (p. 298)

Table 5 Requests categorized by object type (continued)

Object	Requests
NDMP)	- ModifyNDMPConfig (p. 369) - GetNDMPConfigDetails (p. 275)
Quotas	- CreateQuota (p. 154) - ModifyQuota (p. 375) - DeleteQuota (p. 186) - GetAllQuotas (p. 224) - GetQuotaDetails (p. 282)
File Servers	- CreateFileServer (p. 100) - ModifyFileServer (p. 321) - DeleteFileServer (p. 175) - GetAllFileServers (p. 213) - GetFileServerDetails (p. 253) - GetFileServerStatistics (p. 254) - PerformJoinDomain (p. 445)
Secondary network routes for a File Server (optional)	- CreateRoute (p. 157) - DeleteRoute (p. 188) - GetAllRoutes (p. 226) - GetRouteDetails (p. 283)
Secondary virtual network interfaces for a File Server (optional)	- CreateVirtualInterface (p. 164) - DeleteVirtualInterface (p. 192) - GetAllVirtualInterfaces (p. 239) - GetVirtualInterfaceDetails (p. 307)
NIS-alternative files for a File Server (optional)	- GetNISUploadedFilesDetails (p. 277) - PerformNISFileUpload (p. 456) - PerformNISFileDownload (p. 455)
Volume groups	- CreateVolumeGroup (p. 167) - ModifyVolumeGroup (p. 389) - DeleteVolumeGroup (p. 195) - GetAllVolumeGroups (p. 240) - GetVolumeGroupDetails (p. 308)
Volume copies	See: - Snap FSs, filesystem copies, and CloneFSs (p. 34) - Snap LUNs and LUN copies (p. 34)

Table 5 Requests categorized by object type (continued)

Object	Requests
LUNs (see also Snap LUNs and LUN copies on p. 34)	• CreateLUN (p. 135) • DeleteLUN (p. 182) • DeleteVolumeAndChildren (p. 193) • GetAllAvailableLUNNumbers (p. 202) • GetAllLUNs (p. 221) • GetLUNDetails (p. 269) • GetLUNStatistics (p. 272) • GetHostMappingConfigDetails (p. 265) • GetLUNMappingConfigDetails (p. 270) • ModifyLUN (p. 351) • PerformConfigureLUNMapping (p. 412) • PerformRebuildLUNRedundancy (p. 475) • RestoreLUNFromCloneLUN (p. 527)
Snap LUNs and LUN copies	Snap LUN: • CreateSnapLUN (p. 161) • ModifySnapLUN (p. 381) • DeleteSnapLUN (p. 191) • DeleteVolumeAndChildren (p. 193) • GetAllSnapLUNHierarchies (p. 232) • GetAllSnapLUNs (p. 233) • GetSnapLUNDetails (p. 297) LUN duplicate: • PerformLUNDuplication (p. 447) • PerformBackgroundLUNDuplication (p. 402) • PrepareSnapLUN (p. 518) • SyncSnapLUN (p. 529)
Snap FSs, filesystem copies, and CloneFSs	Snap FS: • CreateFileSystemSnapshotSchedule (p. 124) • ModifyFileSystemSnapshot (p. 340) • DeleteFileSystemSnapshot (p. 177) • DeleteFileSystemSnapshotSchedule (p. 179) • DeleteVolumeAndChildren (p. 193) • GetAllFileSystemSnapshots (p. 216) • GetFileSystemSnapshotDetails (p. 256) • GetAllFileSystemSnapshotSchedules (p. 218) • GetFileSystemSnapshotScheduleDetails (p. 258) • PerformRestoreFileSystemFromSnapshot (p. 485)

Table 5 Requests categorized by object type (continued)

Object	Requests
Snap FSs and filesystem copies, continued	Filesystem duplicate: • PerformFileSystemDuplication (p. 425) • PerformBackgroundFileSystemDuplication (p. 400) • PerformRebuildFileSystemRedundancy (p. 474) • PerformRebuildCloneFSRedundancy (p. 473) CloseFS: • CreateCloneFS (p. 97) • DeleteCloneFS (p. 172) • DeleteCloneFS (p. 172) • GetAllCloneFSs (p. 208) • GetCloneFSDetails (p. 245) • ModifyCloneFS (p. 317) •
Schedules (see also Snap FSs, filesystem copies, and CloneFSs on p. 34)	• CreateSchedule (p. 159) • DeleteSchedule (p. 190) • GetAllSchedules (p. 229) • GetScheduleDetails (p. 291) Jobs: • CreateJob (p. 127) • ModifyJob (p. 345) • DeleteJob (p. 181) • GetAllJobs (p. 220) • GetJobDetails (p. 268)
Time management	• ModifyTimeConfig (p. 386) • GetCurrentDateTimeDetails (p. 247) • GetCurrentTimeZoneDetails (p. 248) • GetNTPServerConfigDetails (p. 279)
Administrator accounts	• CreateAccount (p. 92) • ModifyAccount (p. 311) • ModifyMyAccount (p. 367) • DeleteAccount (p. 169) • GetAllAccounts (p. 200) • GetAccountDetails (p. 198) • GetAllSessions (p. 230) Authentication: • Login (p. 309) • Logout (p. 310) • Relogin (p. 524)

Table 5 Requests categorized by object type (continued)

Object	Requests
Notifications and events	• CreateNotificationConnection (p. 151) • CreateNotificationDefinition (p. 152) • ModifyNotificationDefinition (p. 373) • DeleteNotificationDefinition (p. 185) • GetAllNotificationDefinitions (p. 223) • GetNotificationDefinitionDetails (p. 278) • GetAllEvents (p. 210) • PerformTestCallHome (p. 505) • PerformTestEmail (p. 506)
Hardware components (see also Guided Maintenance on p. 46)	• GetProductDetails (p. 281) Pilot: • ModifyManagementConfig (p. 360) • GetPilotDetails (p. 280) Slammers: • ModifySlammer (p. 380) • GetAllSlammers (p. 231) • GetSlammerDetails (p. 295) • GetSlammerStatistics (p. 296) Bricks: • ModifyBrick (p. 314)) • GetAllBricks (p. 204) • GetBrickDetails (p. 242) • GetSpareDriveDetails (p. 299) UPS devices: • CreateExternalUPSDevice (p. 99) • ModifyExternalUPSDevice (p. 319) • DeleteExternalUPSDevice (p. 174) • GetAllExternalUPSDevices (p. 212) • GetExternalUPSDeviceDetails (p. 250) SAN: • GetSANConfigDetails (p. 284) • GetHostMappingConfigDetails (p. 265) • GetAllSANHosts (p. 227) • GetHBAPortConnections (p. 262) • PerformAssociateInitiatorsToHost (p. 398) • DeleteSANHost (p. 189)
Tasks	• GetAllTasks (p. 235) • GetTaskDetails (p. 304) • PerformCancelTask (p. 404)

Table 5 Requests categorized by object type (continued)

Object	Requests
Administrator actions	• DeleteAdministratorAction (p. 170) • GetAllAdministratorActions (p. 201) • GetAdministratorActionDetails (p. 199) • PerformFileSystemCheck (p. 423)
Software updates	• GetAllUpdateComponents (p. 236) • GetAllUpdatePackages (p. 237) • GetUpdatePackageDetails (p. 305) • PerformUpdate (p. 507) • PerformUpdateDiskDrive (p. 509) • PerformUpdateDiskDriveValidation (p. 510) • PerformUpdatePackageStaging (p. 511) • PerformUpdateValidation (p. 513)
Pillar Axiom storage system (system-wide configuration settings)	Monitor: • GetProductDetails (p. 281) • GetSystemStatusDetails (p. 303) • GetCurrentTimeZoneDetails (p. 248) • GetCurrentDateTimeDetails (p. 247) • GetManagementConfigDetails (p. 273) • GetNTPServerConfigDetails (p. 279) • GetNDMPConfigDetails (p. 275) • GetSNMPConfigDetails (p. 298) Modify: • ModifyNDMPConfig (p. 369) • ModifySNMPConfig (p. 384) • ModifyTimeConfig (p. 386) Perform software and firmware updates: • PerformUpdate (p. 507) • PerformUpdateValidation (p. 513)
Sessions	• GetAllSessions (p. 230)
Shut down and restart	• PerformShutdown (p. 495) • PerformRestart (p. 484)

Table 5 Requests categorized by object type (continued)

Object	Requests
Statistics	• GetAllBackupRuns (p. 203) • GetAllSANProtocolStatistics (p. 228) • GetBackupRunStatistics (p. 241) • GetFileServerStatistics (p. 254) • GetFileSystemStatistics (p. 260) • GetHistoricalFileSystemStatistics (p. 263) • GetLUNStatistics (p. 272) • GetProductDetails (p. 281) • GetSANProtocolStatistics (p. 288) • GetSlammerStatistics (p. 296) • PerformCollectSystemInformation (p. 407)
System information (see also Statistics on p. 38)	• GetGUISSLConfigDetails (p. 261) • GetManagementConfigDetails (p. 273) • GetSystemInformationFileDetails (p. 302) • GetSystemStatusDetails (p. 303) • ModifyManagementConfig (p. 360) • PerformCollectSystemInformation (p. 407) • PerformDownloadSystemInformationFile (p. 420) • PerformRestoreFileSystemFromSnapshot (p. 485) System root • PerformResetSystem (p. 482) • PerformUpdateSystemSerialNumber (p. 512)
Tape robot	• GetAllTapeRobots (p. 234) • PerformResetTapeRobotDevice (p. 483) • PerformTapeRobotDiscovery (p. 504)
Tools	• GetAllVerifyOperations (p. 238) • GetDataConsistencyDetails (p. 249) • GetSpareDriveDetails (p. 299) • GetStorageRedundancyDetails (p. 301) • PerformFileSystemCheck (p. 423) • PerformVerifyDataConsistency (p. 515) • PerformVerifyStorageRedundancy (p. 516)

Table 6 lists management requests categorized by the types of actions that you can perform on objects.

Table 6 Requests categorized by action type

Action	Requests
Create	• CreateAccount (p. 92) • CreateCIFSShare (p. 95) • CreateCloneFS (p. 97) • CreateExternalUPSDevice (p. 99) • CreateFileServer (p. 100) • CreateFileSystem (p. 110) • CreateFileSystemSnapshotSchedule (p. 124) • CreateJob (p. 127) • CreateLUN (p. 135) • CreateNFSExport (p. 144) • CreateNFSHostEntry (p. 148) • CreateNotificationConnection (p. 151) • CreateNotificationDefinition (p. 152) • CreateQuota (p. 154) • CreateRoute (p. 157) • CreateSchedule (p. 159) • CreateSnapLUN (p. 161) • CreateVirtualInterface (p. 164) • CreateVolumeGroup (p. 167)

Table 6 Requests categorized by action type (continued)

Action	Requests
Modify	• ModifyAccount (p. 311) • ModifyBrick (p. 314) • ModifyCIFSShare (p. 315) • ModifyCloneFS (p. 317) • ModifyExternalUPSDevice (p. 319) • ModifyFileServer (p. 321) • ModifyFileSystem (p. 330) • ModifyFileSystemSnapshot (p. 340) • ModifyiSCSIPortDetails (p. 343) • PerformAssociateInitiatorsToHost (p. 398) • ModifyJob (p. 345) • ModifyLUN (p. 351) • ModifyManagementConfig (p. 360) • ModifyMyAccount (p. 367) • ModifyNASConfig (p. 368) • ModifyNDMPConfig (p. 369) • ModifyNFSHostEntry (p. 371) • ModifyNotificationDefinition (p. 373) • ModifyQuota (p. 375) • ModifySANHost (p. 377) • ModifySlammer (p. 380) • ModifySnapLUN (p. 381) • ModifySNMPConfig (p. 384) • ModifyTimeConfig (p. 386) • ModifyVolumeGroup (p. 389)

Table 6 Requests categorized by action type (continued)

Action	Requests
Delete	• DeleteAccount (p. 169) • DeleteAdministratorAction (p. 170) • DeleteCIFSShare (p. 171) • DeleteCloneFS (p. 172) • DeleteExternalUPSDevice (p. 174) • DeleteFileServer (p. 175) • DeleteFileSystem (p. 176) • DeleteFileSystemSnapshot (p. 177) • DeleteFileSystemSnapshotSchedule (p. 179) • DeleteJob (p. 181) • DeleteLUN (p. 182) • DeleteNFSExport (p. 183) • DeleteNFSHostEntry (p. 184) • DeleteNotificationDefinition (p. 185) • DeleteQuota (p. 186) • DeleteRoute (p. 188) • DeleteSANHost (p. 189) • DeleteSchedule (p. 190) • DeleteSnapLUN (p. 191) • DeleteVirtualInterface (p. 192) • DeleteVolumeAndChildren (p. 193) • DeleteVolumeGroup (p. 195)

Table 6 Requests categorized by action type (continued)

Action	Requests
Get all	• GetAllAccounts (p. 200) • GetAllAdministratorActions (p. 201) • GetAllAdministratorActions (p. 201) • GetAllBackupRuns (p. 203) • GetAllBricks (p. 204) • GetAllCIFSShares (p. 206) • GetAllCloneFSs (p. 208) • GetAllEvents (p. 210) • GetExternalUPSDeviceDetails (p. 250) • GetAllFileServers (p. 213) • GetAllFileSystems (p. 214) • GetAllFileSystemSnapshots (p. 216) • GetAllFileSystemSnapshotSchedules (p. 218) • GetAllJobs (p. 220) • GetAllLUNs (p. 221) • GetAllNFSExports (p. 222) • GetAllNotificationDefinitions (p. 223) • GetAllQuotas (p. 224) • GetAllQuotaFiles on p. 225 • GetAllRoutes (p. 226) • GetAllSANHosts (p. 227) • GetAllSANProtocolStatistics (p. 228) • GetAllSchedules (p. 229) • GetAllSessions (p. 230) • GetAllSlammers (p. 231) • GetAllSnapLUNHierarchies (p. 232) • GetAllSnapLUNs (p. 233) • GetAllTapeRobots (p. 234) • GetAllTasks (p. 235) • GetAllUpdateComponents (p. 236) • GetAllUpdatePackages (p. 237) • GetAllVerifyOperations (p. 238) • GetAllVirtualInterfaces (p. 239) • GetAllVolumeGroups (p. 240)

Table 6 Requests categorized by action type (continued)

Action	Requests
Get details	• GetAccountDetails (p. 198) • GetAdministratorActionDetails (p. 199) • GetAllCallHomeFileDetails (p. 205) • GetBrickDetails (p. 242) • GetCIFSShareDetails (p. 244) • GetCloneFSDetails (p. 245) • GetCurrentDateTimeDetails (p. 247) • GetCurrentTimeZoneDetails (p. 248) • GetDataConsistencyDetails (p. 249) • GetExternalUPSDeviceDetails (p. 250) • GetFCPortStatistics Details (p. 251) • GetFileServerDetails (p. 253) • GetFileSystemDetails (p. 255) • GetFileSystemSnapshotDetails (p. 256) • GetFileSystemSnapshotScheduleDetails (p. 258) • GetGUISSLConfigDetails (p. 261) • GetHostMappingConfigDetails (p. 265) • GetiSCSIPortStatisticsDetails (p. 266) • GetJobDetails (p. 268) • GetLUNDetails (p. 269) • GetLUNMappingConfigDetails (p. 270) • GetNASConfigDetails (p. 274) • GetNDMPConfigDetails (p. 275) • GetNFSExportDetails (p. 276) • GetNISUploadedFilesDetails (p. 277) • GetNotificationDefinitionDetails (p. 278) • GetNTPServerConfigDetails (p. 279) • GetPilotDetails (p. 280) • GetProductDetails (p. 281) • GetQuotaDetails (p. 282) • GetRouteDetails (p. 283) • GetSANConfigDetails (p. 284) • GetSANHostDetails (p. 285) • GetSANSlammerPortDetails (p. 290) • GetScheduleDetails (p. 291) • GetSlammerControlUnitDiagnosticDetails (p. 293) • GetSlammerDetails (p. 295) • GetSnapLUNDetails (p. 297) • GetSNMPConfigDetails (p. 298) • GetSpareDriveDetails (p. 299) • GetStorageConfig (p. 300)

Table 6 Requests categorized by action type (continued)

Action	Requests
Get details, continued	• GetStorageRedundancyDetails (p. 301) • GetSystemInformationFileDetails (p. 302) • GetSystemStatusDetails (p. 303) • GetTaskDetails (p. 304) • GetUpdatePackageDetails (p. 305) • GetVirtualInterfaceDetails (p. 307) • GetVolumeGroupDetails (p. 308)
Get performance statistics	• GetBackupRunStatistics (p. 241) • GetFileServerStatistics (p. 254) • GetFileSystemStatistics (p. 260) • GetHistoricalFileSystemStatistics (p. 263) • GetLUNStatistics (p. 272) • GetSANProtocolStatistics (p. 288) • GetSlammerStatistics (p. 296)
Get hardware status	• GetPilotDetails (p. 280) • GetSlammerDetails (p. 295) • GetBrickDetails (p. 242) • GetSpareDriveDetails (p. 299) • GetSANConfigDetails (p. 284) • GetSANHostDetails (p. 285) • GetHBAPortConnections (p. 262) • GetSANHostDetails (p. 285) • GetSlammerCommandOutput (p. 292) • GetSlammerControlUnitDiagnosticDetails (p. 293)
Perform (see also Guided Maintenance on p. 46)	• Login (p. 309) • Logout (p. 310) • PerformAcceptAddBrick (p. 391) • PerformAcceptAddSlammer (p. 392) • PerformAcceptBrickDiskDrive (p. 393) • PerformAcceptForeignDiskDrive (p. 395) • PerformAssociateInitiatorsToHost (p. 398) • PerformBackgroundFileSystemDuplication (p. 400)

Table 6 Requests categorized by action type (continued)

Action	Requests
Perform, continued	• PerformBackgroundLUNDuplication (p. 402) • PerformCancelTask (p. 404) • PerformClearLostData (p. 405) • PerformCollectInstantaneousStatistics (p. 406) • PerformCollectSystemInformation (p. 407) • PerformConfigureLUNMapping (p. 412) • PerformCopybackBrickLUNs (p. 415) • PerformDisableSMIStartup (p. 417) • PerformDownloadCallHomeFile (p. 418) • PerformDownloadQuotaFile on p. 419 • PerformDownloadSystemInformationFile (p. 420) • PerformEnableSMIStartup (p. 421) • PerformFileSystemCheck (p. 423) • PerformFileSystemDuplication (p. 425) • PerformIdentifyBrick (p. 433) • PerformIdentifyBrickFRU (p. 435) • PerformIdentifyPilot (p. 437) • PerformIdentifySlammer (p. 438) • PerformIdentifySlammerFRU (p. 440) • PerformIdentifySlammerIOPort (p. 443) • PerformJoinDomain (p. 445) • PerformLUNDuplication (p. 447) • PerformNISFileDownload (p. 455) • PerformNISFileUpload (p. 456) • PerformPeriodicCallHome (p. 460) • PerformPrepareReplaceBrickFRU (p. 461) • PerformPrepareReplaceSlammerFRU (p. 463) • PerformPrepareSlammerMemoryUpgrade (p. 465) • PerformRebuildBrickLUNs (p. 472) • PerformRebuildCloneFSRedundancy (p. 473) • PerformRebuildFileSystemRedundancy (p. 474) • PerformRebuildLUNRedundancy (p. 475) • PerformRejectForeignDiskDrive (p. 478) • PerformRemoveBrickFromConfiguration (p. 480) • PerformRemoveSlammerFromConfiguration (p. 481) • PerformResetSystem (p. 482) • PerformResetTapeRobotDevice (p. 483) • PerformRestart (p. 484) • PerformRestoreFileSystemFromSnapshot (p. 485)

Table 6 Requests categorized by action type (continued)

Action	Requests
Perform, continued	• PerformResumeBrick (p. 486) • PerformResumeBrickFRU (p. 487) • PerformResumeSlammer (p. 489) • PerformResumeSlammerFRU (p. 490) • PerformShutdown (p. 495) • PerformSlammerCommand (p. 496) • PerformSlammerControlUnitDiagnostic (p. 503) • PerformTapeRobotDiscovery (p. 504) • PerformTestCallHome (p. 505) • PerformTestEmail (p. 506) • PerformUpdate (p. 507) • PerformUpdateDiskDrive (p. 509) • PerformUpdateDiskDriveValidation (p. 510) • PerformUpdatePackageStaging (p. 511) • PerformUpdateSystemSerialNumber (p. 512) • PerformUpdateValidation (p. 513) • PerformVerifyDataConsistency (p. 515) • PerformVerifyStorageRedundancy (p. 516) • Relogin (p. 524)
Guided Maintenance	Step 1: Identify hardware component • PerformIdentifyBrick (p. 433) • PerformIdentifyBrickFRU (p. 435) • PerformIdentifyPilot (p. 437) • PerformIdentifySlammer (p. 438) • PerformIdentifySlammerFRU (p. 440) • PerformIdentifySlammerIOPort (p. 443) • GetSlammerCommandOutput (p. 292) • PerformSlammerControlUnitDiagnostic (p. 503) • PerformSlammerControlUnitDiagnostic (p. 503) • PerformSlammerCommand (p. 496) Step 2: Prepare the system • PerformRemoveBrickFromConfiguration (p. 480) • PerformPrepareReplaceBrickFRU (p. 461) • PerformRemoveSlammerFromConfiguration (p. 481) • PerformPrepareSlammerMemoryUpgrade (p. 465)

Table 6 Requests categorized by action type (continued)

Action	Requests
Guided Maintenance, continued	Step 3: Replace hardware component (physical actions to remove old and add new components; no CLI requests) Step 4: Resume system operations • PerformResumeBrick (p. 486) • PerformResumeBrickFRU (p. 487) • PerformResumeSlammer (p. 489) • PerformResumeSlammerFRU (p. 490) Step 5: Verify new hardware component • PerformAcceptAddBrick (p. 391) or • PerformAcceptAddSlammer (p. 392) or • PerformAcceptForeignDiskDrive (p. 395) or PerformRejectForeignDiskDrive (p. 478) Step 6: Rebuild (optional) • PerformRebuildFileSystemRedundancy (p. 474) • PerformRebuildLUNRedundancy (p. 475)

Synchronous and Asynchronous Requests

The CLI supports synchronous and asynchronous requests. Requests that execute quickly (less than 15 seconds) and do not change the system state are generally synchronous requests. These include Get all (p. 42), Get details (p. 43), and some Perform (p. 44) requests. The system simply returns a response, indicating that the request is complete.

Requests that take longer to complete (more than 15 seconds) or change the system state are generally asynchronous requests. These include Create (p. 39), Modify (p. 40), Delete (p. 41), and some Perform (p. 44) requests. With asynchronous requests, the system first checks to see if the request is valid and if the necessary system resources are available. If so, it returns a response with a TaskID and executes the request. You can monitor progress and determine whether the task succeeded or failed using the GetTaskDetails (p. 304) request.

By default, the CLI sends requests synchronously. In addition, there are two request modes:

- Normal mode
- Force-synchronous mode

For request modes in combination with synchronous and asynchronous requests:

- Synchronous requests behave the same way, regardless of which request mode you use.
- Asynchronous requests behave differently, based on request mode:
  - Normal mode tells the Pillar Axiom storage system to return a response when the request is initiated—not when the request has successfully completed.
  - Force-synchronous mode tells the Pillar Axiom storage system to wait until the management request is complete to return a response.

    The force-synchronous mode is the CLI default. Use the PDS_ASYNC (p. 18) configuration variable to send the request in normal mode.

Administrator Roles and Privileges

To administer a Pillar Axiom storage system, you must log in from an administrator account. Every account is assigned a specific role that defines system privileges.

Table 7 Administrator privileges by role

Administrator role	Privileges
Primary system administrator	Perform all configuration, management, and monitoring tasks. This account cannot be deleted or disabled.
Administrator 1	Perform all configuration, management, and monitoring tasks.
Administrator 2	Perform all tasks except: • Create and manage File Servers and administrator accounts. • Modify global, SNMP, and NDMP settings. • Modify software or hardware configurations. • Shut down the system.
Monitor	Display system information only; cannot modify the configuration. Can modify own administrator account attributes.

Table 7 Administrator privileges by role (continued)

Administrator role	Privileges
Support	Perform limited customer service- only functions; cannot modify the configuration. NOTE: Only Pillar World Wide Customer Support Center personnel can use this account.

In order to manage the Pillar Axiom storage system with the CLI, you must have either Primary system administrator or Administrator 1 privileges.

Note: The session timeout may terminate your CLI session during a long-running asynchronous command. The default session timeout is 20 min, which may be too short for some long-running commands. You can extend the session timeout in the GUI using the SessionTimeout parameter of the ModifyManagementConfig request.

System Management Task Summary

Many of the tasks to create Pillar Axiom objects are one-time tasks that you perform as soon as the Pillar Axiom system is installed at your site. Table 8 lists command and request (p. 75) names that you can use to create objects. Follow the checklist sequence to satisfy dependencies among asynchronous commands (see Synchronous and Asynchronous Requests on p. 47).

Consider the Ranges for Option Values (p. 57) as you create objects in the Pillar Axiom system.

- Table 10, Manage command sequence (p. 54), explains how many of each object you can create (minimum to maximum range).
- Table 12, Data type and length ranges (p. 60), explains valid values for the attributes that you define for objects.

Table 8 Create command sequence

Complete?	Required?	Create commands and management requests
□	Required	Log in to the Pillar Axiom storage system: • startsession (p. 79) if using a managed session. Save the session key to use for subsequent commands, which are typically one or more request (p. 75) commands. (For more information, see managed sessions on p. 23 and PDS_SESSIONKEY on p. 18.) • submit if using a set of encapsulated sessions to log in and perform individual requests. (For more information, see encapsulated sessions on p. 23.)
□	Required (some optional substeps)	Configure global system settings and system time: • ModifyManagementConfig (p. 360) enable VLAN Tagging if you want to create more than one File Server • GetNTPServerConfigDetails (p. 279) • GetCurrentDateTimeDetails (p. 247) • GetCurrentTimeZoneDetails (p. 248)
□	Optional	Create volume groups (organizational objects): • GetAllVolumeGroups (p. 240) • CreateVolumeGroup (p. 167)
□	Required (some optional substeps)	Create at least one File Server, or more if you want to segregate network traffic with VLAN Tagging: • CreateFileServer (p. 100) • GetAllFileServers (p. 213) (wait until the status is Available) • (optional) CreateRoute (p. 157) if you need multiple network routes • (optional) CreateVirtualInterface (p. 164) if you need multiple virtual network interfaces • PerformJoinDomain (p. 445) to join a CIFS domain
□	Required (some optional substeps)	Create and configure filesystems: • CreateFileSystem (p. 110) • GetAllFileSystems (p. 214) (wait until the filesystem status is Available) • CreateQuota (p. 154) if you need quota tracking and optional enforcement • Enable and configure filesystem protocols: • CreateCIFSShare (p. 95) • CreateNFSExport (p. 144) • GetAllNFSExports (p. 222) (wait until the status is Available) • CreateNFSHostEntry (p. 148)

Table 8 Create command sequence (continued)

Complete?	Required?	Create commands and management requests
□	Optional	Create alerts and notifications: • CreateNotificationConnection (p. 151) • CreateNotificationDefinition (p. 152)
□	Optional	Enable and configure SNMP: • GetSNMPConfigDetails (p. 298) • ModifySNMPConfig (p. 384)
□	Optional	Enable and configure NDMP: • GetNDMPConfigDetails (p. 275) • ModifyNDMPConfig (p. 369)
□	Optional	Schedule data replication: • CreateSchedule (p. 159) • GetAllSchedules (p. 229) to retrieve the schedule's unique identifier for the next step • CreateJob
□	Optional	Create additional administrator accounts: • GetAllAccounts (p. 200) • CreateAccount (p. 92)
□	Optional	Rename hardware components to meaningful names: • ModifyBrick (p. 314) • ModifySlammer (p. 380) • ModifyExternalUPSDevice (p. 319) • PerformAssociateInitiatorsToHost (p. 398)
□	Optional	Log out if necessary to end a managed session: endsession (p. 68)

Monitor Objects

All administrator roles can monitor the system status and display configuration details about Pillar Axiom objects. Table 9 is a checklist of command and request (p. 75) names that you can use as a starting point to monitor the Pillar Axiom storage system.

Table 9 Monitor command sequence

Complete?	Required?	Monitor commands and management requests
□	Required	Log in to the CLI: • startsession (p. 79) if using a managed session. Save the session key to use for subsequent commands, which are typically one or more request (p. 75) commands. (For more information, see managed sessions on p. 23 and PDS_SESSIONKEY on p. 18.) • submit (p. 82) if using a set of encapsulated sessions to log in and perform individual requests. (For more information, see encapsulated sessions on p. 23.)
□	Optional	Monitor the system status and health: • GetSystemStatusDetails (p. 303) • GetAllTasks (p. 235) • GetAllNotificationDefinitions (p. 223) • GetAllEvents (p. 210) • GetAllAdministratorActions (p. 201)
□	Optional	Monitor volume groups: • GetAllVolumeGroups (p. 240) • GetVolumeGroupDetails (p. 308) • GetAllLUNs (p. 221) • GetLUNDetails (p. 269)
□	Optional	Monitor File Servers: • GetAllFileServers (p. 213) • GetFileServerDetails (p. 253) • GetFileServerStatistics (p. 254) • GetAllRoutes (p. 226) • GetRouteDetails (p. 283) • GetAllVirtualInterfaces (p. 239) • GetVirtualInterfaceDetails (p. 307)

Table 9 Monitor command sequence (continued)

Complete?	Required?	Monitor commands and management requests
□	Optional	Monitor filesystems: • GetAllFileSystems (p. 214) • GetFileSystemDetails (p. 255) • GetFileSystemStatistics (p. 260) • GetHistoricalFileSystemStatistics (p. 263) • GetAllCIFSShares (p. 206) • GetCIFSShareDetails (p. 244) • GetAllNFSExports (p. 222) • GetNFSExportDetails (p. 276) • GetAllQuotas (p. 224) • GetQuotaDetails (p. 282)
□	Optional	Monitor SNMP and NDMP configurations: • GetSNMPConfigDetails (p. 298) • GetNDMPConfigDetails (p. 275)
□	Optional	Monitor data replicas and replication schedules: • GetAllFileSystemSnapshots (p. 216) and GetFileSystemSnapshotDetails (p. 256) • GetAllFileSystemSnapshotSchedules (p. 218) and GetFileSystemSnapshotScheduleDetails (p. 258) • GetAllSnapLUNs (p. 233), GetAllSnapLUNHierarchies (p. 232), and GetAllSnapLUNs (p. 233)
□	Optional	Monitor administrator accounts: • GetAllAccounts (p. 200) • GetAccountDetails (p. 198) • GetAllSessions (p. 230) • GetAdministratorActionDetails (p. 199)
□	Optional	Monitor software versions and installed features: • GetAllUpdateComponents (p. 236) • GetAllUpdatePackages (p. 237) • GetUpdatePackageDetails (p. 305)

Table 9 Monitor command sequence (continued)

Complete?	Required?	Monitor commands and management requests
□	Optional	Monitor hardware components: • GetAllBricks (p. 204) • GetBrickDetails (p. 242) • GetSpareDriveDetails (p. 299) • GetPilotDetails (p. 280) • GetAllSlammers (p. 231) • GetSlammerDetails (p. 295) • GetSlammerStatistics (p. 296) • GetUpdatePackageDetails (p. 305)\ • GetSANConfigDetails (p. 284) • GetHostMappingConfigDetails (p. 265) • GetAllSANHosts (p. 227) • GetHBAPortConnections (p. 262)
□	Optional	Log out if necessary to end a managed session: endsession (p. 68)

Manage Objects

Table 10, Manage command sequence, is a checklist of command and request (p. 75) names that you can use to modify or delete Pillar Axiom objects.

Note: You must delete child objects that are lower in a hierarchical structure before you can delete the parent objects. For example, you cannot delete a volume group if filesystems are associated with the volume group.

Table 10 Manage command sequence

Complete?	Required?	Manage commands and management requests
□	Required	Log in to the CLI: • startsession (p. 79) if using a managed session. Save the session key to use for subsequent commands, which are typically one or more request (p. 75) commands. (For more information, see managed sessions on p. 23 and PDS_SESSIONKEY on p. 18.) • submit (p. 82) if using a set of encapsulated sessions to log in and perform individual requests. (For more information, see encapsulated sessions on p. 23.)

Table 10 Manage command sequence (continued)

Complete?	Required?	Manage commands and management requests
□	Optional	Modify or delete filesystems: • ModifyFileSystem (p. 330) • ModifyCIFSShare (p. 315) • DeleteCIFSShare (p. 171) • ModifyNFSHostEntry (p. 371) • DeleteNFSHostEntry (p. 184) • DeleteNFSExport (p. 183) • ModifyQuota (p. 375) • DeleteQuota (p. 186) • DeleteFileSystem (p. 176)
□	Optional	Modify or delete File Servers: • ModifyFileServer (p. 321) • DeleteRoute (p. 188) • DeleteVirtualInterface (p. 192) • DeleteFileServer (p. 175)
□	Optional	Modify or delete volume groups: • ModifyVolumeGroup (p. 389) • DeleteVolumeGroup (p. 195) • ModifyLUN (p. 351) • DeleteLUN (p. 182)
□	Optional	Modify or delete notifications (alerts): • ModifyNotificationDefinition (p. 373) • DeleteNotificationDefinition (p. 185)
□	Optional	Delete data replication schedules as needed: • CreateFileSystemSnapshotSchedule (p. 124). Note: You cannot modify a replication schedule through the CLI. You must create a new schedule and delete the old one. (See CreateFileSystemSnapshotSchedule on p. 124 and DeleteFileSystemSnapshotSchedule on p. 179)
□	Optional	Create data replicas as needed: • PerformLUNDuplication (p. 447)
□	Optional	Modify or delete administrator accounts: • ModifyAccount (p. 311) • ModifyMyAccount (p. 367) • DeleteAccount (p. 169)

Table 10 Manage command sequence (continued)

Complete?	Required?	Manage commands and management requests
□	Optional	Modify hardware component names: • ModifyManagementConfig (p. 360) • ModifyBrick (p. 314) • ModifySlammer (p. 380) • ModifyExternalUPSDevice (p. 319) • PerformAssociateInitiatorsToHost (p. 398)
□	Optional	Perform software upgrades: • GetAllUpdatePackages (p. 237) • GetAllUpdateComponents (p. 236) • GetUpdatePackageDetails (p. 305) • PerformUpdatePackageStaging (p. 511) • PerformUpdate (p. 507) • PerformUpdateValidation (p. 513) • PerformUpdateDiskDrive (p. 509) • PerformUpdateDiskDriveValidation (p. 510)
□	Optional	Perform support functions as needed: • DeleteExternalUPSDevice (p. 174) • PerformFileSystemCheck (p. 423) • PerformRestoreFileSystemFromSnapshot (p. 485) • PerformRebuildFileSystemRedundancy (p. 474) • PerformRebuildLUNRedundancy (p. 475) • PerformVerifyDataConsistency (p. 515) and GetDataConsistencyDetails (p. 249) • GetSpareDriveDetails (p. 299) • PerformVerifyStorageRedundancy (p. 516) and GetStorageRedundancyDetails (p. 301) • GetSystemInformationFileDetails (p. 302) • PerformShutdown (p. 495) and PerformRestart (p. 484) With an encrypted file from Pillar World Wide Customer Support Center (because user data and configuration information are deleted): • PerformResetSystem (p. 482) • PerformUpdateSystemSerialNumber (p. 512)

Table 10 Manage command sequence (continued)

Complete?	Required?	Manage commands and management requests
□	Optional	Perform Guided Maintenance as needed. • Identify the hardware component: • PerformIdentifyBrick (p. 433) • PerformIdentifyBrickFRU (p. 435) • PerformIdentifySlammer (p. 438) • PerformIdentifySlammerFRU (p. 440) • PerformIdentifySlammerIOPort (p. 443) • Prepare the system: • PerformPrepareReplaceBrickFRU (p. 461) • PerformPrepareReplaceSlammerFRU (p. 463) • PerformPrepareSlammerMemoryUpgrade (p. 465) • Replace hardware components (physical actions to remove old and add new). • Resume system operations: • PerformResumeBrick (p. 486) • PerformResumeBrickFRU (p. 487) • PerformResumeSlammer (p. 489) • PerformResumeSlammerFRU (p. 490) • Verify the new hardware component: • PerformAcceptAddBrick (p. 391) • PerformAcceptAddSlammer (p. 392) • PerformAcceptForeignDiskDrive (p. 395) or PerformRejectForeignDiskDrive (p. 478)
□	Optional	Log out if necessary to end a managed session: endsession (p. 68)

Ranges for Option Values

Remember the following ranges as you create, modify, and delete Pillar Axiom objects to manage a Pillar Axiom system.

- Table 11, Quantity ranges, defines quantity ranges.
- Table 12, Data type and length ranges, defines additional ranges.

Table 11 Quantity ranges

Object	Quantity range
File Servers	Maximum = 8 for each NAS Slammer. Note: In multi-Slammer systems, virtual interfaces (VIFs) that are associated with a File Server can be configured on multiple Slammers. The presence of VIFs is what counts against the limit. Such a File Server is considered to be present on each Slammer on which it has VIFs. Note: Virtual local area network (VLAN) tagging does not need to be enabled for more than one File Server. If VLAN tagging is enabled, File Servers do not require a unique VLAN tag.
VLANs	• Minimum = 0 • Maximum = 32
Virtual interfaces for each File Server	• Minimum = 1 (primary interface) • Maximum = 16 for each port, 32 total (primary plus up to 31 secondary)
Network routes for each File Server	• Minimum = 0 • Maximum (default) = 8 • Maximum (static) = 16
Configuration files (NIS alternative files)	Maximum = 50 MB Note: Size limit for each NIS file (/etc/passwd, /etc/group, and /etc/netgroup) that is uploaded to the Pilot.
Upload file size	Maximum = 650 MB for a software update package.
Volume groups	• Minimum = 1 (root /Volumes) • Maximum = 5000 • 100 nested volume groups within each group • 4 levels of nested volume groups
Filesystems	• Minimum = 1 • Maximum = 1024
Filesystem size	• Minimum = 1 to 2 GB. The exact value depends on these factors: • Brick type (Fibre Channel or SATA) • RAID geometry (RAID 5 or Distributed RAID) • Strip size (1 MB or normal) • Maximum = system capacity Note: All capacity values must be in increments of 1 GB.

Table 11 Quantity ranges (continued)

Object	Quantity range
SecureWORMfs Retention Period	• Default = 0 days - 1000 years and must be greater than or equal to the minimum and less than or equal to the maximum retention value. • Minimum = 0 days -1000 years. • Maximum = 0 days - 1000 years and must be greater than or equal to the minimum retention value.
NFS exports	Maximum = 1000 for each File Server
NFS host entries	Maximum = 4000 for each File Server
CIFS shares	Maximum = 128 for each File Server
CIFS client connections	Maximum for each NAS Slammer (specified memory is the total combined memory of both control units): • 1200 for 12 GB total memory • 6000 for 24 GB total memory • 12,000 for 48 GB total memory
CIFS security groups	Maximum = 1024 for each user
Snap FS	Maximum = 250 Snap FSs for each filesystem; 16,000 Snap FSs total for each Pillar Axiom system
SAN LUNs	Maximum = • 4096 visible for each Pillar Axiom system (1024 if all LUNs have nonzero clone repository) • 255 visible for each host • 4096 visible for each SAN Slammer Note: A visible (active) SAN LUN requires one virtual LUN (VLUN). A clone of a SAN LUN requires a VLUN for the metadata and another for the data repository. If that clone is active, a third VLUN is required, making a total of four VLUNs for the SAN LUN and its clone.
SAN LUN size	• Minimum: 1 to 2 GB. The exact value depends on these factors • Brick type (Fibre Channel or SATA) • RAID geometry (RAID 5 or Distributed RAID) • Strip size (1 MB or normal) • Maximum: System capacity Note: All capacity values must be in increments of 1 GB.
Pillar Axiom Path Manager (SAN host software)	Maximum Pillar Axiom systems = 8 for each SAN host
Pillar Axiom Path Manager data paths	Maximum = 32 to each LUN

Table 11 Quantity ranges (continued)

Object	Quantity range
Pillar Axiom Path Manager FC HBA ports	Maximum = 32 for each SAN host
Clone LUNs	Maximum = • Number of available LUNs • 13 active at a time (of a single source)
iSCSI	Maximum = • 256 TCP connections for each iSCSI port • 256 iSCSI initiators for each iSCSI port • 32 persistent reservation registration keys for each LUN • 512 simultaneous commands for each iSCSI port
Administrator accounts	• Minimum = 2 (for the Primary system administrator and Support Administrator) • Maximum = 23
Administrator sessions	Maximum = 10 total at any given time (2 reserved for Primary system administrator and Support Administrator)
NDMP sessions	Maximum concurrent backup and restore sessions for each system = 10

Table 12 Data type and length ranges

Field	Length / Type	Notes
Names for: • Alerts • Brick storage enclosures • File Servers • Filesystems • Pillar Axiom system • Schedules • Slammer storage controllers • Volume groups	1 through 16 8-bit Unicode Transformation Format (UTF-8) printable characters. UTF-8 is described in RFC 2279, which you can find online with any internet search engine	Embedded spaces are permitted. Invalid characters: • Nonprintable characters, including ASCII 0 through 31 • / (slash) and \ (backslash) • . and .. (dot and dot-dot alone) • Embedded tabs Pillar Axiom processing: • Leading and trailing white space is stripped • Comparison is case sensitive Tip: Names of filesystems that you export to NFS users should contain only US-ASCII characters. Note: You can have filesystems with the same name if the filesystems are not in the same volume group or File Server.

Table 12 Data type and length ranges (continued)

Field	Length / Type	Notes
LUN names	1 through 82 UTF-8 printable characters	Invalid characters: • Nonprintable characters, including ASCII 0 through 31 • / (slash) and \ (backslash) • . and .. (dot and dot-dot alone) • Embedded tabs
Host names	1 through 63 UTF-8 printable characters	
Names for: • DNS domains • NIS domains	1 through 256 printable characters	
Snap FS name	1 through 26 UTF-8 printable characters	Invalid characters: • / (slash) and \ (backslash) • . and .. (dot and dot-dot alone) • Embedded tabs
Snap FS base (mount) name	8 through 33 UTF-8 printable characters	Invalid characters: • / (slash) and \ (backslash) • . and .. (dot and dot-dot alone) • Embedded tabs
Administrator user name	1 through 16 UTF-8 printable characters	Case-sensitive value Invalid characters: • Embedded spaces • / (slash)
Administrator password	6 through 16 UTF-8 printable characters	• Case-sensitive value • Embedded spaces are permitted.
Administrator login attempts	1 through 20 (integer)	
Optional entries for administrator full names	0 through 40 UTF-8 printable characters	Embedded spaces are permitted.
Optional entries for telephone numbers	0 through 80 UTF-8 printable characters	Embedded spaces are permitted.
Alert descriptions	0 through 80 UTF-8 printable characters	Embedded spaces are permitted.

Table 12 Data type and length ranges (continued)

Field	Length / Type	Notes
Email address (emailuser@host)	1 through 64 characters for email user	a-z A-Z 0-9 ! # $ % & ' * + - / = ? ^ _ ` { | } ~ . are permitted, except that . (dot) cannot be the first or last character.
	1 through 255 characters for host	a-z A-Z 0-9 - . are permitted, except that: • 0-9 - . cannot be the first character. • . - cannot be the last character. An IP address cannot be the host part of the email address.
NDMP account user name	1 through 16 UTF-8 printable characters	Case-sensitive value Invalid characters: • Embedded spaces • / (slash)
NDMP account password	6 through 8 ASCII printable characters	Case-sensitive value
Names for CIFS: • Servers • Domains	1 through 15 ASCII printable characters 33 through 126	
Comments for CIFS servers	1 through 44 ASCII printable characters 32 through 126	Embedded spaces are permitted.
Names for CIFS shares	1 through 80 ASCII printable characters 32 through 126	Embedded spaces are permitted. Invalid characters: • / (slash) and \ (backslash) • : (colon) • control character
Comments for CIFS shares	0 through 256 ASCII printable characters 32 through 126	Embedded spaces are permitted.
CIFS administrator (for Domain Controller): • User name • Password	0 through 256 UTF-8 characters	Case-sensitive value Invalid characters: • Embedded spaces • / (slash)
Directory paths for CIFS shares	1024 bytes and start with a \ (backslash)	Path includes the filesystem name of up to 40 printable characters, plus a NULL terminator.

Table 12 Data type and length ranges (continued)

Field	Length / Type	Notes
CIFS security group names	0 through 322 ASCII printable characters 32 through 126	Invalid characters: • Embedded tabs, carriage returns, and newlines • ?, *, <, >, +, =, comma(,), |, [,]; :, ;, ", \, ', /
Directory paths for NFS exports	UTF–8 characters up to 1024 bytes in length; start with a / (slash)	Path includes a filesystem name, which can consist of up to 40 UTF-8 printable characters, plus a NULL terminator.
NFS host name	UTF–8 characters up to 255 bytes in length	Host format: • IP address in dotted-decimal format • Subnet address with both the subnet and mask in dotted-decimal format • Host name • Asterisk (*), to export to all NFS clients (everyone)
IP addresses	0 through 255, in all four parts	IP version 4 (IPv4) dotted-decimal notation (xxx.xxx.xxx.xxx)
Virtual LAN (VLAN) ID (tag)	0 through 4094 (integer)	• 1 through 4094 denote that VLAN Tagging is enabled. • 0 denotes that VLAN Tagging is disabled.
SNMP community string	0 through 255 ASCII printable characters 33 through 126	Invalid characters: • Embedded spaces • Control characters
Chap Secrets	100 UTF-8 characters	Non-character (for example, integer) CHAP secret values are not supported. CHAP secrets should be more than 12 bytes if IPsec is not used on insecure network segments.

CHAPTER 3

CLI Commands

This chapter describes the CLI command set. (See Chapter 4 on p. 90 for command requests.) Please note:

- The command SYNTAX and OPTIONS sections show both long and short option names. EXAMPLES show the more descriptive long names.
- You can shorten a command name to a three-letter abbreviation. For example, you can shorten the request (p. 75) command name to `req`.
- OPTIONS are not order-dependent; however, if you use the `{-C | --config}` option, it must appear first in the command string.

Command Reference Pages

The CLI commands are listed in alphabetic order.

NAME **convert**

DESCRIPTION Converts the specified input to a different output format. This command is intended for Pillar Customer Service and advanced users.

SYNTAX

```
pdscli convert
   [{-C | --config} <file>]
   [{-W | --tmpdir} <directory>]
   [{-f | --force}]
   [{-x | --extract} <extract-field>]
   [{-o | --outputform} {form | msg | pretty |
     raw | cli | none}]
   {<request-name> |
     {{-m | --msg} <xml-msg-file> |
     {-b | --body} <xml-body-file> |
     {-t | --template} <template-file>}
   [<name>=<val> | @ <file>]...
```

OPTIONS

[{-C | --config} <ConfigFile>]

You may enter the name of a configuration file that contains a set of command options. If you use this option, it must appear first in the command string.

[{-W | --tmpdir} <directory>]

Optionally enter the directory where the CLI should write any temporary files it needs to use while processing this request.

[{-f | --force}]

Optionally specify {-f | --force} to send the message even if the message is invalid.

[{-x | --extract} <extract-field>]

Optionally enter a slash-separated path through the XML document to an extract field. The command interpreter parses the XML document until it finds an extract field that matches the pattern and displays the field on the standard output device (stdout).

The CLI tracing option is very useful for displaying the XML output for extract field.

[{-o | --outputform} {form | msg | pretty | raw | cli | none}]

Optionally enter a value that identifies how to display the output that the Pillar Axiom storage system generates when you run a command or request option. Choose from:

- `form` so that the response is parsed and displayed as a list of `Name: <value>` lines.

- `msg` so that the response is parsed and displayed as a continuous string of XML tags for all of the fields in the response.
  Note: Although the output is in XML format, it is not formatted to send directly to the server.
- `pretty` so that the response is parsed and displayed as a series of hierarchically indented lines, with each XML tag on a separate line.
- `raw` so that the response is displayed exactly as received. The response is not parsed.
- `cli` so that the response is parsed and displayed as an input template. Use this option with the `convert` command.
- `none` so that the output format is neither parsed nor displayed.

If you do not specify this value, the Pillar Axiom storage system uses the default `form` value.

<request-name>

Enter a case-sensitive request name. A request is composed of one or more field/value pairs. Generate a template so that you have a complete set of field/value pairs for a request.

To display a list of valid request names, enter the following command:

```
pdscli help list-requests
```

{-m | --msg} <xml-msg-file>

Enter the name of an input file that contains a complete and valid XML message, normally MCCRequest. MCC stands for Management and Control Console. MCCRequest and the corresponding MCCResponse are available in the CLI’s supplemental xml subdirectory (see “Extract the pdscli Executable and Supplemental Files” on page 16 for more information).

{-b | --body} <xml-body-file>

Enter the name of an input file that contains a complete and valid XML request body. The `pdscli` inserts the request body into the appropriate XML document type to send to the Pillar Axiom storage system.

{-t | --template} <template-file>

Enter the name of an input file that contains a CLI request template. The first line of the file must be:

```
Request=<RequestName>
```

The `Request=<RequestName>` line is followed by a series of `<name>=<value>` lines that supply the request options. Enter <values> that are appropriate for your storage environment.

[<name=val>]...

If the {<request-name>} includes options, you may enter a case-sensitive <name>=<value> pair and repeat as necessary.

[@<file>]...

If the {<request-name>} includes options, you may specify an input file that contains case-sensitive <name>=<value> pairs. You may repeat this argument as necessary.

EXAMPLES

You have started to investigate using XML for data exchange and would like to convert a PerformDownloadSystemInformationFile template to XML format:

```
pdscli convert --outputform msg
--template PerformDownloadSystemInformationFile.txt
```

SEE ALSO

CLI configuration variables on p. 17

submit on p. 82

template on p. 86

verify on p. 88

Ranges for Option Values on p. 57

NAME **endsession**

DESCRIPTION Terminates a command line interface (CLI) session. Use this command in combination with the `startsession` and `request` commands, for example, to optimize system performance while running scripts.

SYNTAX

```
pdscli endsession
   [{-C | --config} <file>]
   [{-W | --tmpdir} <directory>]
   [{-H | --host} <host-name>]
   [{-T | --timeout} <timeout-sec>]
   [{-A | --async}]
   [{-s | --session} <sessionID>]
   [{-x | --extract} <extract-field>]
   [{-o | --outputform} {form | msg | pretty
      | raw | cli | none}]
```

OPTIONS

[{-C | --config} <ConfigFile>]

You may enter the name of a configuration file that contains a set of command options. If you use this option, it must appear first in the command string.

[{-W | --tmpdir} <directory>]

Optionally enter the directory where the CLI should write any temporary files it needs to use while processing this request.

[{-H | --host} <host-name>]

Enter the name or management IP address of the Pilot in the Pillar Axiom storage system.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

[{-T | --timeout} <timeout-sec>]

Enter an integer that specifies an inactivity time limit, in seconds, after which the connection is terminated.

[{-A | --async}]

Set this variable if you want the system to return asynchronous responses instead of waiting for the request to complete.

[{-s | --session} <sessionID>]

Enter the unique identifier of the administrative session to end. The system returnes this value when you sent the `startsession` command

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

[{-x | --extract} <extract-field>]

Optionally enter a slash-separated path through the XML document to an extract field. The command interpreter parses the XML document until it finds an extract field that matches the pattern and displays the field on the standard output device (stdout).

The CLI tracing option is very useful for displaying the XML output for extract field.

[{-o | --outputform} {form | msg | pretty | raw | cli | none}]

Optionally enter a value that identifies how to display the output that the Pillar Axiom storage system generates when you run a command or request option. Choose from:

- `form` so that the response is parsed and displayed as a list of `Name: <value>` lines.
- `msg` so that the response is parsed and displayed as a continuous string of XML tags for all of the fields in the response.
  Note: Although the output is in XML format, it is not formatted to send directly to the server.
- `pretty` so that the response is parsed and displayed as a series of hierarchically indented lines, with each XML tag on a separate line.
- `raw` so that the response is displayed exactly as received. The response is not parsed.
- `cli` so that the response is parsed and displayed as an input template. Use this option with the `convert` command.
- `none` so that the output format is neither parsed nor displayed.

If you do not specify this value, the Pillar Axiom storage system uses the default `form` value.

EXAMPLES

Run the following command to end the current administrative session.

```
pdscli endsession
   -s aeea1234abc5678efab1234abcd1234
```

where the long string is the session ID with the startsession request.

SEE ALSO

CLI configuration variables on p. 17

startsession on p. 79

Ranges for Option Values on p. 57

NAME **help**

DESCRIPTION Displays help text about the command line interface (CLI) or the specified <item>.

SYNTAX

```
pdscli help
pdscli help <command>
pdscli help options
pdscli help environment
pdscli help list-requests
pdscli help template -d <request>
pdscli help list-errors
```

OPTIONS

<command>

Displays help text for the specified <command>:

- convert
- endsession
- rawsend
- request
- startsession
- submit
- template
- verify

options

Displays a list of frequently used command options. (See also “CLI configuration variables” on page 17.)

environment

Displays a list of environment variable values. (See also “Set Configuration Variables with Environment Variables” on page 19.)

list-requests

Displays a list of <request-names> that you can specify as a <request-name> for the request, submit, template, and verify commands. (See also “Pillar Axiom Objects and Management Requests” on page 31.)

template [-d] <request>

Displays individual request information.

list-errors

Displays a list of error messages that you might encounter as you work from the CLI.

EXAMPLES Run the following command to retrieve a list of valid configuration requests.

```
pdscli help list-requests > CLI-requests.txt
```

help list-requests | sort [sort options] > requests.txt will generate an alphabetic sorted list of requests as defined by the sort utility of your operating system.

The system writes the entire list of requests to a text file. You can refer to the file as you plan how to implement and manage your Pillar Axiom storage system. (The file contents are not included here to conserve space.)

SEE ALSO

convert on p. 65

endsession on p. 68

rawsend on p. 72

request on p. 75

startsession on p. 79

submit on p. 82

template on p. 86

verify on p. 88

NAME **rawsend**

DESCRIPTION Adds a required header to the message. This command is intended for Pillar Customer Service and advanced users.

SYNTAX

```
pdscli rawsend
   [{-C | --config} <file>]
   [{-W | --tmpdir} <directory>]
   [{-H | --host} <host-name>]
   [{-T | --timeout} <timeout-sec>]
   [{-A | --async}]
   {{-m | --msg} <xml-msg-file> |
   {-b | --body} <xml-body-file> |
   {-t | --template} <template-file>}}
   [{-f | --force}]
   [{-x | --extract} <extract-field>]
   [{-o | --outputform} <{form | msg | pretty
     | raw | cli | none}>]
   {<XML-file>}
```

OPTIONS

[{-C | --config} <ConfigFile>]

You may enter the name of a configuration file that contains a set of command options. If you use this option, it must appear first in the command string.

[{-W | --tmpdir} <directory>]

Optionally enter the directory where the CLI should write any temporary files it needs to use while processing this request.

[{-H | --host} <host-name>]

Enter the name or management IP address of the Pilot in the Pillar Axiom storage system.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

[{-T | --timeout} <timeout-sec>]

Enter an integer that specifies an inactivity time limit, in seconds, after which the connection is terminated.

[{-A | --async}]

Set this variable if you want the system to return asynchronous responses instead of waiting for the request to complete.

[{-m | --msg} <xml-msg-file>]

Enter the name of an input file that contains a complete and valid XML message, normally MCCRequest. MCC stands for Management and Control Console. MCCRequest and the corresponding MCCResponse are available in the CLI's supplemental xml subdirectory (see "Extract the pdscli Executable and Supplemental Files" on page 16 for more information).

[{-b | --body} <xml-body-file>]

Enter the name of an input file that contains a complete and valid XML request body. The `pdscli` inserts the request body into the appropriate XML document type to send to the Pillar Axiom storage system.

[{-t | --template} <template-file>]

Enter the name of an input file that contains a CLI request template. The first line of the file must be:

```
Request=<RequestName>
```

The `Request=<RequestName>` line is followed by a series of `<name>=<value>` lines that supply the request options. Enter <values> that are appropriate for your storage environment.

[{-f | --force}]

Optionally specify {-f | --force} to send the message even if the message is invalid.

[{-x | --extract} <extract-field>]

Optionally enter a slash-separated path through the XML document to an extract field. The command interpreter parses the XML document until it finds an extract field that matches the pattern and displays the field on the standard output device (stdout).

The CLI tracing option is very useful for displaying the XML output for extract field.

[{-o | --outputform} <{form | msg | pretty | raw | cli | none}>]

Optionally enter a value that identifies how to display the output that the Pillar Axiom storage system generates when you run a command or request option. Choose from:

- `form` so that the response is parsed and displayed as a list of `Name: <value>` lines.
- `msg` so that the response is parsed and displayed as a continuous string of XML tags for all of the fields in the response.
  Note: Although the output is in XML format, it is not formatted to send directly to the server.
- `pretty` so that the response is parsed and displayed as a series of hierarchically indented lines, with each XML tag on a separate line.
- `raw` so that the response is displayed exactly as received. The response is not parsed.
- `cli` so that the response is parsed and displayed as an input template. Use this option with the `convert` command.

- `none` so that the output format is neither parsed nor displayed.

If you do not specify this value, the Pillar Axiom storage system uses the default `form` value.

{<XML-file>}

Enter the name of a file that contains the XML message to send. The Pillar Axiom storage system does not validate the file contents.

EXAMPLES

You have started to investigate using XML for data exchange. Use the `rawsend` command to experiment in your XML investigations.

SEE ALSO

CLI configuration variables on p. 17

startsession on p. 79

help on p. 70

endsession on p. 68

Synchronous and Asynchronous Requests on p. 47

Ranges for Option Values on p. 57

NAME **request**

DESCRIPTION Sends a command that performs the specified <request-name> within a managed session. Use this command in combination with the `startsession` and `endsession` commands, for example, to optimize system performance while running scripts.

SYNTAX

```
pdscli request
   [{-C | --config} <file>]
   [{-W | --tmpdir} <directory>]
   [{-H | --host} <host-name>]
   [{-T | --timeout} <timeout-sec>]
   [{-A | --async}]
   [{-s | --session} <sessionID>]
   [{-f | --force}]
   [{-x | --extract} <extract-field>]
   [{-o | --outputform} <{form | msg | pretty
      | raw | cli | none}>]
   {<request-name> |
     {-m | --msg} <xml-msg-file> |
     {-b | --body} <xml-body-file> |
     {-t | --template} <template-file>}
   [<name>=<val> | @ <file>]...
```

OPTIONS

[{-C | --config} <ConfigFile>]

You may enter the name of a configuration file that contains a set of command options. If you use this option, it must appear first in the command string.

[{-W | --tmpdir} <directory>]

Optionally enter the directory where the CLI should write any temporary files it needs to use while processing this request.

[{-H | --host} <host-name>]

Enter the name or management IP address of the Pilot in the Pillar Axiom storage system.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

[{-T | --timeout} <timeout-sec>]

Enter an integer that specifies an inactivity time limit, in seconds, after which the connection is terminated.

[{-A | --async}]

Set this variable if you want the system to return asynchronous responses instead of waiting for the request to complete.

[{-s | --session} <sessionID>]

Enter the unique session ID, which is a 32-character value that consists of any ASCII-printable characters except a space. Use the `startsession` command to get the unique session ID.

Example session ID:

```
4J616d44604D537973206278976F642E
```

[{-f | --force}]

Optionally specify {-f | --force} to send the message even if the message is invalid.

[{-x | --extract} <extract-field>]

Optionally enter a slash-separated path through the XML document to an extract field. The command interpreter parses the XML document until it finds an extract field that matches the pattern and displays the field on the standard output device (stdout).

The CLI tracing option is very useful for displaying the XML output for extract field.

[{-o | --outputform} <{>form | msg | pretty | raw | cli | none}]

Optionally enter a value that identifies how to display the output that the Pillar Axiom storage system generates when you run a command or request option. Choose from:

- `form` so that the response is parsed and displayed as a list of `Name: <value>` lines.
- `msg` so that the response is parsed and displayed as a continuous string of XML tags for all of the fields in the response.
  Note: Although the output is in XML format, it is not formatted to send directly to the server.
- `pretty` so that the response is parsed and displayed as a series of hierarchically indented lines, with each XML tag on a separate line.
- `raw` so that the response is displayed exactly as received. The response is not parsed.
- `cli` so that the response is parsed and displayed as an input template. Use this option with the `convert` command.
- `none` so that the output format is neither parsed nor displayed.

If you do not specify this value, the Pillar Axiom storage system uses the default `form` value.

<request-name>

Enter a case-sensitive request name. A request is composed of one or more field/value pairs. Generate a template so that you have a complete set of field/value pairs for a request.

To display a list of valid request names, enter the following command:

```
pdscli help list-requests
```

{-m | --msg} <xml-msg-file>

Enter the name of an input file that contains a complete and valid XML message, normally MCCRequest. MCC stands for Management and Control Console. MCCRequest and the corresponding MCCResponse are available in the CLI's supplemental xml subdirectory (see "Extract the pdscli Executable and Supplemental Files" on page 16 for more information).

{-b | --body} <xml-body-file>

Enter the name of an input file that contains a complete and valid XML request body. The `pdscli` inserts the request body into the appropriate XML document type to send to the Pillar Axiom storage system.

{-t | --template} <template-file>

Enter the name of an input file that contains a CLI request template. The first line of the file must be:

```
Request=<RequestName>
```

The `Request=<RequestName>` line is followed by a series of `<name>=<value>` lines that supply the request options. Enter <values> that are appropriate for your storage environment.

[<name=val>]...

If the {<request-name>} includes options, you may enter a case-sensitive <name>=<value> pair and repeat as necessary.

[@<file>]...

If the {<request-name>} includes options, you may specify an input file that contains case-sensitive <name>=<value> pairs. You may repeat this argument as necessary.

EXAMPLES

You run the following command to start an administrative session.

```
pdscli startsession --host storage.xyz.corp
```

You are prompted to enter your user name and password because you did not specify those options. When the Pillar Axiom storage system returns a session ID, you copy the value from the command prompt.

You open your previously defined `pdscli .cfg` configuration file and update the value of the PDS_SESSIONKEY environment variable with the session ID that you copied from the command prompt.

You run the following command to retrieve system status details.

```
pdscli request --host 172.18.64.107 --session
4J616d44604D537973206278976F642E GetSystemStatusDetails
```

SEE ALSO

CLI configuration variables on p. 17

Management Requests on p. 90

startsession on p. 79

endsession on p. 68

submit on p. 82

template on p. 86

verify on p. 88

Synchronous and Asynchronous Requests on p. 47

PDS_SESSIONKEY on p. 18

Ranges for Option Values on p. 57

NAME **startsession**

DESCRIPTION Starts a command line interface (CLI) session as an alternative to the `submit` command. The `submit` command automatically starts a session, sends a request, and ends a session for each request; the `startsession` command, however, allows you send multiple requests without closing the session each time. Use this command in combination with the `request` and `endsession` commands, for example, to optimize system performance while running scripts.

Tip: You can use the Unix shell of your choice or a Windows Command Line to store the SessionID as an environment variable; however, this is not as reliable as storing the SessionID in a temp file.

The session timeout may terminate your CLI session during a long-running asynchronous command. The default session timeout is 20 minutes, which may be too short for some long-running commands. You can extend the session timeout in the GUI using the SessionTimeout parameter of the ModifyManagementConfig request.

SYNTAX

```
pdscli startsession
   [{-C | --config} <file>]
   [{-W | --tmpdir} <directory>]
   [{-H | --host} <host-name>]
   [{-T | --timeout} <timeout-sec>]
   [{-A | --async}]
   [{-u | --user} <user-name>]
   [{-p | --password} <password>]
   [{-x | --extract} <extract-field>]
   [{-o | --outputform} <{form | msg | pretty
      | raw | cli | none}>]
```

OPTIONS

[{-C | --config} <ConfigFile>]

Enter the name of a configuration file that contains a set of command options. If used, this option must be first in the command string.

[{-W | --tmpdir} <directory>]

Optionally enter the directory where the CLI should write any temporary files it needs to use while processing this request.

[{-H | --host} <host-name>]

Enter the name or management IP address of the Pilot in the Pillar Axiom storage system.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

[{-T | --timeout} <timeout-sec>]

Enter an integer that specifies an inactivity time limit, in seconds, after which the connection is terminated.

[{-A | --async}]

Set this variable if you want the system to return asynchronous responses instead of waiting for the request to complete.

[{-u | --user} <user-name>]

Enter a valid user name that is defined for a Pillar Axiom administrator account.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

[{-p | --password} <password>]

Enter a valid password that is defined for the specified user name.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

[{-x | --extract} <extract-field>]

Optionally enter a slash-separated path through the XML document to an extract field. The command interpreter parses the XML document until it finds an extract field that matches the pattern and displays the field on the standard output device (stdout).

The CLI tracing option is very useful for displaying the XML output for extract field.

[{-o | --outputform} <{form | msg | pretty | raw | cli | none}>]

Optionally enter a value that identifies how to display the output that the Pillar Axiom storage system generates when you run a command or request option. Choose from:

- `form` so that the response is parsed and displayed as a list of `Name: <value>` lines.
- `msg` so that the response is parsed and displayed as a continuous string of XML tags for all of the fields in the response.
  Note: Although the output is in XML format, it is not formatted to send directly to the server.
- `pretty` so that the response is parsed and displayed as a series of hierarchically indented lines, with each XML tag on a separate line.
- `raw` so that the response is displayed exactly as received. The response is not parsed.
- `cli` so that the response is parsed and displayed as an input template. Use this option with the `convert` command.
- `none` so that the output format is neither parsed nor displayed.

If you do not specify this value, the Pillar Axiom storage system uses the default `form` value.

EXAMPLES

Run the following command to start an administrative session at the command prompt.

```
pdscli startsession --host storage.xyz.corp
```

You are prompted to enter your user name and password because you did not specify those options. Save the sessionID value that the system returns; you will need this for the `request` and `endsession` commands.

SEE ALSO

CLI configuration variables on p. 17

request on p. 75

endsession on p. 68

help on p. 70

Synchronous and Asynchronous Requests on p. 47

Ranges for Option Values on p. 57

NAME **submit**

DESCRIPTION Logs in to the CLI, sends the specified <request-name> in the requested format, displays the response in the requested output format, and logs out.

SYNTAX

```
pdscli submit
   [{-C | --config} <file>]
   [{-W | --tmpdir} <directory>]
   [{-H | --host} <host-name> ]
   [{-T | --timeout} <timeout-sec>]
   [{-A | --async}]
   [{-u| --user} <user-name> ]
   [{-p| --password} <password>]
   [{-f | --force}]
   [{-x | --extract} <extract-field>]
   [{-o | --outputform} <{form | msg | pretty
     | raw | cli | none}>]
   {<request-name> |
     {{-m | --msg} <xml-msg-file> |
     {-b | --body} <xml-body-file> |
     {-t | --template} <template-file>}
```

OPTIONS

[{-C | --config} <ConfigFile>]

Enter the name of a configuration file that contains a set of command options. If used, this option must be first in the command string.

[{-W | --tmpdir} <directory>]

Optionally enter the directory where the CLI should write any temporary files it needs to use while processing this request.

[{-H | --host} <host-name>]

Enter the name or management IP address of the Pilot in the Pillar Axiom storage system.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

[{-T | --timeout} <timeout-sec>]

Enter an integer that specifies an inactivity time limit, in seconds, after which the connection is terminated.

[{-A | --async}]

Set this variable if you want the system to return asynchronous responses instead of waiting for the request to complete.

[{-u| --user} <user-name>]

Enter a valid user name that is defined for a Pillar Axiom administrator account.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

[{-p| --password} <password>]

Enter a valid password that is defined for the specified user name.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

[{-f | --force}]

Optionally specify {-f | --force} to send the message even if the message is invalid.

[{-x | --extract} <extract-field>]

Optionally enter a slash-separated path through the XML document to an extract field. The command interpreter parses the XML document until it finds an extract field that matches the pattern and displays the field on the standard output device (stdout).

The CLI tracing option is very useful for displaying the XML output for extract field.

[{-o | --outputform} <{form | msg | pretty | raw | cli | none}>]

Optionally enter a value that identifies how to display the output that the Pillar Axiom storage system generates when you run a command or request option. Choose from:

- `form` so that the response is parsed and displayed as a list of `Name: <value>` lines.
- `msg` so that the response is parsed and displayed as a continuous string of XML tags for all of the fields in the response.
  Note: Although the output is in XML format, it is not formatted to send directly to the server.
- `pretty` so that the response is parsed and displayed as a series of hierarchically indented lines, with each XML tag on a separate line.
- `raw` so that the response is displayed exactly as received. The response is not parsed.
- `cli` so that the response is parsed and displayed as an input template. Use this option with the `convert` command.
- `none` so that the output format is neither parsed nor displayed.

If you do not specify this value, the Pillar Axiom storage system uses the default `form` value.

<request-name>

Enter a case-sensitive request name. A request is composed of one or more field/value pairs. Generate a template so that you have a complete set of field/value pairs for a request.

To display a list of valid request names, enter the following command:

```
pdscli help list-requests
```

{-m | --msg} <xml-msg-file>

Enter the name of an input file that contains a complete and valid XML message, normally MCCRequest. MCC stands for Management and Control Console. MCCRequest and the corresponding MCCResponse are available in the CLI's supplemental xml subdirectory (see "Extract the pdscli Executable and Supplemental Files" on page 16 for more information).

{-b | --body} <xml-body-file>

Enter the name of an input file that contains a complete and valid XML request body. The `pdscli` inserts the request body into the appropriate XML document type to send to the Pillar Axiom storage system.

{-t | --template} <template-file>

Enter the name of an input file that contains a CLI request template. The first line of the file must be:

```
Request=<RequestName>
```

The `Request=<RequestName>` line is followed by a series of `<name>=<value>` lines that supply the request options. Enter <values> that are appropriate for your storage environment.

EXAMPLES

Run the following command to check the status of your Pillar Axiom storage system without starting and stopping a session to run several commands.

```
pdscli submit -H 172.18.64.107 -u Xavier
   -p s7Z12W79 GetSystemStatusDetails
```

The system returns the following command results (edited to conserve space).

```
MCCResponse:
   ...
   GetSystemStatusDetailsResponse:
     SystemName:                        XYZ-data
     SystemStatus:                      Normal
   ..
```

SEE ALSO

CLI configuration variables on p. 17

Management Requests on p. 90

request on p. 75

template on p. 86

verify on p. 88

Synchronous and Asynchronous Requests on p. 47

Ranges for Option Values on p. 57

NAME **template**

DESCRIPTION Creates a template for a <request-name>, which is an action to perform on an object.

SYNTAX

```
pdscli template
   [{-C | --config} <file>]
   [{-W | --tmpdir} <directory>]
   [{-d | --description} ]
   {<request-name>}
```

OPTIONS

[{-C | --config} <ConfigFile>]

Enter the name of a configuration file that contains a set of command options. If used, this option must be first in the command string.

[{-W | --tmpdir} <directory>]

Optionally enter the directory where the CLI should write any temporary files it needs to use while processing this request.

[{-d | --description}]

Displays descriptive text about request options, if descriptive text is defined.

{<request-name>}

Enter a case-sensitive request name. A request is composed of one or more field/value pairs. Generate a template so that you have a complete set of field/value pairs for a request.

To display a list of valid request names, enter the following command:

```
pdscli help list-requests
```

EXAMPLES Run the following command to create a template for the CreateAccount request. This streamlined template does not contain the descriptions that are included in the provided `CreateAccountRequest.req` file.

```
pdscli template CreateAccount >
   CreateAccount.template
```

The `CreateAccount.template` is as follows.

```
Request=CreateAccount
Username       =
Fullname       =
Password       =
RetypePassword =
EmailAddress   =
PhoneNumber    =
Status         =
Role           =
```

Enter the following values in the template to create an administrator account for Jim Smith.

```
Request=CreateAccount
Username       =jsmith
Fullname       =Jim Smith
Password       =ChangeMe1st
RetypePassword =ChangeMe1st
EmailAddress   =jimsmith@xyz.com
PhoneNumber    =223.555.1235
Status         =Enabled
Role           =Type1
```

SEE ALSO

CLI configuration variables on p. 17

Management Requests on p. 90

request on p. 75

submit on p. 82

verify on p. 88

Ranges for Option Values on p. 57

NAME **verify**

DESCRIPTION Verifies command input for validity. The command interpreter runs this command on your client workstation and does not submit anything to the Pillar Axiom storage system. You can optionally submit the request after you verify its contents with this command.

SYNTAX

```
pdscli verify
   [{-W | --tmpdir} <directory>]
   {<request-name> |
     {-m | --msg} <xml-msg-file> |
     {-b | --body} <xml-body-file> |
     {-t | --template} <template-file>}
   [<name>=<val> | @ <file>]...
```

OPTIONS

[{-W | --tmpdir} <directory>]

Optionally enter the directory where the CLI should write any temporary files it needs to use while processing this request.

{<request-name>}

Enter a case-sensitive request name. A request is composed of one or more field/value pairs. Generate a template so that you have a complete set of field/value pairs for a request.

To display a list of valid request names, enter the following command:

```
pdscli help list-requests
```

{-m | --msg} <xml-msg-file>

Enter the name of an input file that contains a complete and valid XML message, normally MCCRequest. MCC stands for Management and Control Console. MCCRequest and the corresponding MCCResponse are available in the CLI's supplemental xml subdirectory (see “Extract the pdscli Executable and Supplemental Files” on page 16 for more information).

{-b | --body} <xml-body-file>

Enter the name of an input file that contains a complete and valid XML request body. The `pdscli` inserts the request body into the appropriate XML document type to send to the Pillar Axiom storage system.

{-t | --template} <template-file>

Enter the name of an input file that contains a CLI request template. The first line of the file must be:

```
Request=<RequestName>
```

The `Request=<RequestName>` line is followed by a series of `<name>=<value>` lines that supply the request options. Enter <values> that are appropriate for your storage environment.

[<name=val>]...

If the {<request-name>} includes options, you may enter a case-sensitive <name>=<value> pair and repeat as necessary.

[@<file>]...

If the {<request-name>} includes options, you may specify an input file that contains case-sensitive <name>=<value> pairs. You may repeat this argument as necessary.

EXAMPLES

Run the following command to verify that the `CreateAccount.template` file is a complete and valid CLI template to be used as an input file.

```
pdscli verify CreateAccount
   -t CreateAccount.template
```

You have installed Slammer field replaceable units (FRUs), and you want to ensure that the Slammer1 and Slammer2 are still valid names for the GetSlammerDetails (p. 295) request before you Use Scripts to Run Commands (p. 21):

```
pdscli verify GetSlammerDetails
   Slammer=/Company-XZY-data/Slammer1
   Slammer=/Company-XZY-data/Slammer2
```

SEE ALSO

CLI configuration variables on p. 17

Management Requests on p. 90

request on p. 75

submit on p. 82

template on p. 86

Ranges for Option Values on p. 57

CHAPTER 4

Management Requests

Use management requests to administer the Pillar Axiom storage system. Run either of the following commands to transmit management requests from the CLI client software on your administrative workstation to the Pilot management controller:

```
pdscli request (p. 75) <request-name>
or
pdscli submit (p. 82) <login-options> <request-name>
```

The management requests are documented in the same format as the commands in CLI Commands (p. 64). The reference pages also point to the output files that show example response data. (See Extract the pdscli Executable and Supplemental Files on p. 16 for more information.)

Note: If you use the sample output files to create your own code, omit any extraneous colons. Colons should separate fields only.

Request Reference Pages

Management requests are listed in alphabetic order.

NAME **CancelForceFileSystemRebalance**

DESCRIPTION Removes the flag that the ForceFileSystemRebalanceOnRestart request sets.

SYNTAX

```
pdscli {request | submit} CancelForceFileSystemRebalance
```

OPTIONS None

EXAMPLES Run the following command to force the reassignment of filesystems across all NAS Slammer CUs.

```
pdscli request CancelForceFileSystemRebalance
```

See the CancelForceFileSystemRebalance.txt file for example response data.

SEE ALSO ForceFileSystemRebalanceOnRestart on p. 196

NAME **CreateAccount**

DESCRIPTION Creates a new administrator account. The role that you assign to the account determines the administrator's privileges in a Pillar Axiom storage system. Only Type 1 and Primary Administrator accounts can create new adminstrator accounts.

SYNTAX

```
pdscli {request | submit} CreateAccount
   Username=login-name
   Fullname="First Last"
   Password=password
   RetypePassword=password
   EmailAddress=name@domain.com
   PhoneNumber=telephone-number
   Status={Enabled | Disabled}
   Role={Type1 | Type2 | Type3}
   SSHState={Enabled | Disabled}
```

OPTIONS

Username

Identifies the administrator's login (user) name.

Fullname [optional]

Identifies the administrator's full name ("first" and "last").

Password

Identifies the login password. Passwords are case sensitive, and blank passwords are not permitted.

RetypePassword

Enter the password again.

EmailAddress [optional]

Identifies the recipient's email address. The email server to which the Pillar Axiom system sends alerts must be able to send messages to this email address.

PhoneNumber [optional]

Identifies the recipient's telephone number. The Pillar Axiom system does not verify the validity of this entry.

Note: Enclose telephone numbers with spaces in quotes.

Status

Identifies whether the administrator account is enabled.

- Enabled accounts are active. Administrators whose accounts are enabled can log in to the Pillar Axiom storage system.
- Disabled accounts are inactive. Administrators whose accounts are disabled cannot log in.

Role

Identifies the administrator’s role. A role defines the administrator’s privileges. Choose from:

- Type1, if the person can perform all configuration and administration tasks.
- Type2, if the person can perform all tasks except create, modify, and delete administrator accounts and File Servers; modify global, SNMP, and NDMP settings; modify software or hardware configurations; or shut down the system.
- Type3, if the person can display information only, and cannot modify the configuration.

You cannot assign administrators to the following predefined roles:

- PrimarySystem, which identifies the Primary system administrator. This role has the same privileges as the Type1 role.
- PrimarySupport, which identifies the Primary Support Administrator. This role can perform support-related activities.

You cannot delete the PrimarySystem and PrimarySupport accounts.

SSHState [optional]

Identifies whether the support personnel can SSH to the Pilot controller.

- Enabled permits Pillar support personnel to run an SSH session.
- Disabled prohibits running an SSH session.

This flag does not allow the administrator to SSH to the Pilot.

Only Administrator Type 1 accounts can enable or disable the SSHState flag.

⚠ CAUTION

To avoid system corruption, use extreme caution when running an SSH session. It is recommended that you use the ModifyAccount request whenever possible and keep SSH disabled.

EXAMPLES

Run the following command to create a Pillar Axiom account for Yvette Smith.

```
pdscli request CreateAccount @input
```

where input.req contains the following name/value pairs:

```
Request=CreateAccount
Username=Yvette
Fullname=Yvette Smith
Password=ChangeMe3
RetypePassword=ChangeMe3
EmailAddress=yvette@xyz.com
PhoneNumber=800-555-1235
Status=Enabled
```

```
Role=Type1
SSHState=Disabled
```

See the CreateAccountResponse.txt file for example response data.

SEE ALSO

ModifyAccount on p. 311

ModifyMyAccount on p. 367

DeleteAccount on p. 169

GetAccountDetails on p. 198

GetAllAccounts on p. 200

Ranges for Option Values on p. 57

NAME **CreateCIFSShare**

DESCRIPTION Creates a Common Internet File System (CIFS) share for a filesystem. A CIFS share is a filesystem mount point for CIFS clients (users).

SYNTAX

```
pdscli {request | submit} CreateCIFSShare
  Name=share-name
  Comment=CIFS Share Comment
  FileSystem=GUID-or-FQN-of-filesystem
  Enabled={true | false}
  Path=path-to-directory
```

OPTIONS

Name

Identifies the name of a CIFS share. A filesystem must be shared before users can create or access data files. Share names must be unique within a File Server.

Comment [optional]

Describes the CIFS share.

FileSystem

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the FQN of the filesystem with which this share is associated. If there is more than one file server, use the FQN that includes the file server and filesystem.

Tip: To create an administrative share for a file system, create the filesystem$ share. For example, for a file system named `foo`, create the `foo$` share.

Tip: You can use the GetAllFileSystems request to determine the GUID.

Enabled

Identifies whether the CIFS share is enabled.

- Enabled shares are active. Users can access an enabled share point.
- Disabled shares are inactive. Users cannot access a disabled share point.

Path

Enter the full path to the CIFS share.

EXAMPLES

Run the following command to create a share for your engineering organization’s filesystem.

```
pdscli request CreateCIFSShare @input
```

where input.req contains the following name/value pairs:

```
Request=CreateCIFSShare
Name=EngShare
Comment=Share for Engineering
FileSystem=/Company-XYZ-FS1/EngAtl
Enabled=true
Path=/src
```

See the CreateCIFSShareResponse.txt file for example response data.

SEE ALSO

ModifyCIFSShare on p. 315

DeleteCIFSShare on p. 171

GetAllFileSystems on p. 214

GetAllCIFSShares on p. 206

GetCIFSShareDetails on p. 244

Ranges for Option Values on p. 57

NAME **CreateCloneFS**

DESCRIPTION A Clone FS is a point-in-time, writeable snapshot of a filesystem, which can be used later to revert the filesystem. This snapshot retains the same QoS parameters as the source filesystem. It consumes storage capacity from the Clone FS storage space created for the source filesystem. A Clone FS can be scheduled to occur at any time. This provides a convenient method to branch from the source data without the need to do a full block-level copy.

SYNTAX

```
pdscli {request | submit} CreateCloneFS
   SourceFileSystemFQN=FQN-of-source-filesystem
   FileServerFQN=FQN-of-file-server
   FileSystemName=name-of-filesystem
   Active={true | false}
   EnforceVolumeRetention{true | false}
```

OPTIONS

SourceFileSystemFQN

Identifies a specific Pillar Axiom object.

- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

  FQN Format:
  `/Fully/Qualified/ObjectName`

  Enter the FQN of the filesystem with which this share is associated. If there is more than one file server, use the FQN that includes the file server and filesystem.

FileServerFQN

Enter the FQN of a File Server from which the new filesystem gets protocol-access options.

FileSystemName

Identifies the name that is assigned to a filesystem. Each filesystem name must be unique within its associated File Server and volume group. You must specify the File Server and the filesystem name, which results in a Fully Qualified File System name of /fileserver/filesystemname.

Active [optional]

Identifies whether the CloneFS is available (active) or is not seen (inactive) when it is created. Inactive CloneFSs can be changed to active; however, once the Clone FS is set to active, you cannot change it to inactive. Default is active.

RetentionPolicy.EnforceVolumeRetention

Enforces the retention policy settings for a filesystem.

Files written over CIFS connections are protected immediately, not allowing any further modifications.

Files written over NFS connections must be protected explicitly by setting the Immutable field for the file in its corresponding attribute file.

For instructions on how to edit the attribute file, refer to the *Pillar Axiom Administrator's Guide*.

EXAMPLES

Run the following command to create an active Clone FS named Company-XYZ-CFS1. This example assumes that one File Server exists.

```
pdscli request CreateCloneFS @input
```

where input.req contains the following name/value pairs:

```
Request=CreateCloneFS
FileServerFQN=/Company-XYZ-CFS1
FileSystemName=EngAtl
Active=true
```

See the CreateCloneFSResponse.txt file for example response data.

SEE ALSO

GetCloneFSDetails on p. 245

GetAllSnapLUNs on p. 233

DeleteCloneFS on p. 172

GetAllCloneFSHierarchies on p. 207

GetAllCloneFSs on p. 208

NAME

CreateExternalUPSDevice

DESCRIPTION

Identifies a new uninterruptable power supply (UPS) device for the system to monitor. You may connect several UPS devices to the system. The system retrieves information from each device using SNMP over the Pilot ethernet connection to monitor and report status.

SYNTAX

```
pdscli {request | submit} CreateExternalUPSDevice
   IPAddress=IP-address
   SNMPCommunityString=string
```

OPTIONS

IPAddress

Identifies the IP address that is assigned to the External UPS Device.

SNMPCommunityString

Identifies a community for which a specific trap host should receive traps that the Pillar Axiom system generates. You can specify different community strings for each trap host so that multiple administrators can receive specific types of SNMP traps.

Default: public (note the lower-case spelling)

EXAMPLES

Run the following command to create an external UPS device.

```
pdscli request CreateExternalUPSDevice @input
```

where input.req contains the following name/value pairs:

```
Request=CreateExternalUPSDevice
IPAddress=172.18.65.128
SNMPCommunityString=public
```

See the CreateExternalUPSDeviceResponse.txt file for example response data.

SEE ALSO

DeleteExternalUPSDevice on p. 174

GetAllExternalUPSDevices on p. 212

GetExternalUPSDeviceDetails on p. 250

ModifyExternalUPSDevice on p. 319

SNAME **CreateFileServer**

DESCRIPTION Creates a File Server in a Pillar Axiom storage system. A File Server is defined as:

> A NAS-only object that is assigned security, network, and protocol access attributes. The attributes apply to all filesystems that are associated with that specific File Server. A Pillar Axiom NAS system requires at least one File Server.

SYNTAX

```
pdscli {request | submit} CreateFileServer
  Name=File-Server-name
  Comment=File-Server-Comment
  PrimaryVirtualInterface.IPAddress=
    IP-address
  PrimaryVirtualInterface.IPNetmask=
    IP-netmask
  PrimaryVirtualInterface.VLANTag=
    {integer-1-through-4095}
  PrimaryVirtualInterface.FrameSize=integer
  PrimaryVirtualInterface.
      NetworkInterfaceIOPort.Slammer=
      GUID-or-FQN-of-Slammer
  PrimaryVirtualInterface.
    NetworkInterfaceIOPort.ControlUnitNumber
    ={0 | 1}
  PrimaryVirtualInterface.
    NetworkInterfaceIOPort.EthernetPortName=
    {PORT0 | PORT1}
  DefaultRoute.GatewayIPAddress=IP-address
  DNSConfig.DomainName=string
  DNSConfig.Server[1-3]IPAddress=IP-address
  NISConfig.DomainName=string
  NISConfig.ServerIPAddress=IP-address
  NSSSearchOrder.HostLookupService[1-2]=
    {NIS | DNS | None}
  NSSSearchOrder.PasswordLookupService[1-2]=
    {NIS | Files | None}
  AccountMapping.MapIdenticalNames=
    {true | false}
  AccountMapping.UseAllDomains=
    {true | false}
  AccountMapping.DomainName=string
  CIFSServer.ServerName=CIFS-server-name
  CIFSServer.Comment=CIFS-server-comment
  CIFSServer.CharacterSet={Standard-ASCII |
    IBM-437 | IBM-850}
```

```
CIFSServer.OpportunisticLockingEnabled=
  {true | false}
CIFSServer.WINSServer[1-3]IPAddress=
  IP-address
CIFSServer.DomainName=string
CIFSServer.UseAnonymousPDCAccess=
  {true | false}
CIFSServer.PDCAccessAccount.DomainName=
  string
CIFSServer.PDCAccessAccount.Username=
  string
CIFSServer.PDCAccessAccount.Password=
  string
NFSServer.PortNumber=integer
NFSServer.CharacterSet={Standard-ASCII |
  UTF8 | ISO8859-1}
NFSServer.AnonymousUID=integer-negative-OK
NFSServer.AnonymousGID=integer-negative-OK
NFSServer.MountReservePortOnly=
  {true | false}
NFSServer.AllowNonRootChown={true | false}
NFSServer.TCPSupported={true | false}
NFSServer.TCPConnectionsLimit=integer
NFSCharacterSet={Standard-ASCII |
  UTF8 | ISO8859-1}
```

OPTIONS

Name

Identifies the name that is assigned to a File Server. File Server names must be unique on the Pillar Axiom storage system.

Comment

Describes the File Server.

PrimaryVirtualInterface.IPAddress

Identifies the IP address that is assigned to the primary virtual network interface.

PrimaryVirtualInterface.IPNetmask

Identifies the subnet mask that is assigned to the File Server’s virtual network interface.

PrimaryVirtualInterface.VLANTag

Identifies the virtual LAN (VLAN) ID that is assigned to the File Server’s virtual network interface.

- Assign a value of 1 through 4094, inclusive, if you have connected a VLAN-capable switch to the Pillar Axiom system.
- Leave the field blank to disable VLAN tagging.

PrimaryVirtualInterface.FrameSize

Identifies the packet frame size. This value defines the maximum transmission unit (MTU).

The frame size (MTU) does not include the Ethernet header portion of the packet. If your network switch has trouble with this, you can set the switch to a larger value or lower the MTU size to correct the problem.

- If your network supports extended Ethernet (jumbo) frames, enter an integer greater than 1500 and less than 16362. Make sure that this Pillar Axiom MTU size matches the network MTU size. If the MTU sizes are mismatched, users may experience I/O “hangs” when the client machines try to process too-large packets.
- If your network does not support jumbo frames, enter the default frame size of 1500.

PrimaryVirtualInterface.NetworkInterfaceIOPort.Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer that is the File Server’s primary virtual interface; for example, /Slammer1.

PrimaryVirtualInterface.NetworkInterfaceIOPort.ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

PrimaryVirtualInterface.NetworkInterfaceIOPort.EthernetPortName

Identifies the port on the Slammer that is assigned to the File Server’s virtual network interface. Select a gigabit Ethernet network port on on one of the data-path Slammers control unit (CU) .

- If Link Aggregation is enabled, select port 0 (zero). You cannot create virtual interfaces on alternate port 1 when Link Aggregation is enabled.
- If Link Aggregation is disabled, select either port.

DefaultRoute.GatewayIPAddress

Identifies the IP address that is assigned to the gateway host. The gateway IP address is used to route messages from this network to other networks.

DNSConfig.DomainName [optional]

Identifies a Domain Name Service (DNS) domain, which specifies the domain to be searched.

DNSConfig.Server[1-3]IPAddress [optional]

Identifies the IP address of the Domain Name Service (DNS) server in your network that translates host names to IP addresses.

NISConfig.DomainName [optional]

Identifies a Network Information Service (NIS) domain, which is a group of computers that share the same configuration information.

NISConfig.ServerIPAddress [optional]

Identifies the IP address of a Network Information Service (NIS) server that holds a database of user accounts (among other NIS-related entries). When a user attempts to access the Pillar Axiom system, the user is authenticated against the database.

NSSSearchOrder.HostLookupService[1-2] AND NSSSearchOrder.PasswordLookupService[1-2]

Identifies the search order that the Pillar Axiom system uses to resolve hosts and passwords. Choose from:

- NIS (if you use Network Information Services), DNS (if you use Domain Name System), or None to resolve hosts.
- NIS (if you use Network Information Services), Files (if you use /etc/passwd, /etc/group, and
/etc/netgroup files), or None to resolve passwords.

If you want the system to search only one source for host resolution, password resolution, or both, enter None for the second search target.

AccountMapping.MapIdenticalNames

Identifies whether support for mapping CIFS-to-NFS user accounts is enabled.

- Enable CIFS-to-NFS mapping if you have a single authority to assign users with both an NFS account and CIFS account using the same name. Users can use NFS only, CIFS only, or both NFS and CIFS.
- Disable CIFS-to-NFS mapping if you don't have a single authority to assign NFS and CIFS accounts.

You should disable account mapping when:

- Users do not have both NFS and CIFS accounts.
- NFS and CIFS users account names are different.
- The authority to assign account names can be done by different groups.

Note that the user’s NFS account is the owner of files and directories that the user creates.

AccountMapping.UseAllDomains

Identifies whether CIFS-to-NFS account mapping is enabled for all CIFS domains.

- Enable this option to use CIFS-to-NFS account mapping for all CIFS domains.
- Enable this option to use CIFS-to-NFS account mapping for a single CIFS domain that you specify.

AccountMapping.DomainName

Identifies the CIFS domain name. Enter the NetBIOS name of the CIFS domain that contains definitions for authentication and account mapping.

Enter a CIFS domain name when you enable MapIdenticalNames and disable UseAllDomains.

CIFSServer.ServerName [optional]

Identifies the name by which users refer to the CIFS server. The CIFS server advertises its services by this name through the NetBIOS Naming Service, so this value is sometimes called the NetBIOS name.

CIFSServer.Comment [optional]

Describes the CIFS server. Clients who browse or list shares may view these comments.

CIFSServer.CharacterSet [optional]

Identifies which character set is valid. Choose from:

- Standard-ASCII, which is a 7-bit character set that includes 128 printable and control character values (0-127).
- IBM-437, which is an 8-bit character set that includes the Standard-ASCII values plus values from OEM code page 437. Used on most PCs that are sold in the USA.
- IBM-850, which is an 8-bit character set that includes the Standard-ASCII values plus values from OEM code page 850. Used on most PCs that are sold in Western European countries.

CIFSServer.OpportunisticLockingEnabled [optional]

Identifies whether opportunistic locks (oplocks) are enabled to cache client data.

- Read caching permits data that is stored on the server to be copied to the client. When a client application requests the data, the read request is satisfied by reading the client’s copy of the data. Read caching can last forever. If the data that is stored on the server changes, the client is informed and the read cache is invalidated.
- Write caching permits data changes to be buffered in the client for up to two minutes. When a client application changes the data, the write request is satisfied by changing the client’s copy of the data. Client-server connection problems that occur during the two-minute window can result in lost changes. If the client is still running, a popup message alerts the client to the lost changes.

Disable oplocks to require that all client requests must access a data file instead of cache.

CIFSServer.WINSServer[1-3]IPAddress [optional]

Identifies a Windows Internet Naming Service (WINS) server in your network. The WINS server is the central repository for Windows server names. Broadcasts of NetBIOS names do not cross subnets, so the WINS server can resolve names across multiple subnets.

Note: If you enable Active Directory for authentication, you do not need to enter a WINS server address.

CIFSServer.DomainName [optional]

Identifies the CIFS domain name. Enter the NetBIOS name of the CIFS domain that contains definitions for authentication and account mapping.

CIFSServer.UseAnonymousPDCAccess [optional]

Identifies whether the File Server communicates with the CIFS Domain Controller as an anonymous user to authenticate users and to perform user-name-to-SID mappings (SID = security identifier).

- Enable this option if your Domain Controller is installed on a pre-Windows 2000 server or a Windows 2000 server that is a member of a pre-Windows 2000 domain. These configurations permit access by anonymous users.
- Disable this option if your Domain Controller is installed on a Windows Server 2003 machine. That configuration denies access by anonymous users.

CIFSServer.PDCAccessAccount.PDCName [optional]

Identifies the name of a Windows Domain Controller that maintains a database of user, group, and machine accounts. Enter the Domain Controller’s name. Ensure that the Domain Controller is registered with the WINS or DNS servers that you specify. The Pillar Axiom system uses WINS or DNS, in that order, to resolve the name and locate the Domain Controller.

In this release, you must log in to a Windows 2000 native-mode domain controller as an administrator, run the following command, and reboot the domain controller:

```
net localgroup "Pre-Windows 2000 Compatible Access" everyone /add
```

This command relaxes a security restriction in Windows 2000 servers to disallow anonymous access to user lists and groups.

If you use a Windows 2000 mixed-mode domain controller for CIFS authentication, you do not have to run the command and reboot.

CIFSServer.PDCAccessAccount.Username [optional]

Identifies the administrator user name for the Domain Controller that is associated with the Pillar Axiom storage system.

CIFSServer.PDCAccessAccount.Password [optional]

Identifies the administrator password for the CIFS Domain Controller.

NFSServer.PortNumber [optional]

Identifies which port is reserved for NFS requests. If you allow NFS requests only from a secure port, enter a port number that is less than or equal to 1024.

Default: port 2049

NFSServer.CharacterSet [optional]

Identifies which character set is valid. Choose from:

- Standard-ASCII, which is a 7-bit character set that includes 128 printable and control character values (0-127).
- UTF8, which is a Unicode-coded character set that uses an 8-bit transformation format. Defined in RFC 2279.
- ISO8859-1, which is an 8-bit, single-byte character set that is an extension of Standard-ASCII.

NFSServer.AnonymousUID [optional]

Identifies the user ID (UID) for anonymous users.

- Set the UID to zero (0) if you want anonymous-user access to be identified as the root user. Be aware, however, that this is a very insecure setting.
- Set the UID to the “nobody” value (often -2, but not always) if you want anonymous-user access to be identified as the user “nobody.” In most instances, this is the recommended setting.
- Set the UID to any other user ID that you want to be identified as anonymous-user access.

NFSServer.AnonymousGID [optional]

Identifies the group ID (GID) of a file where the group owner is not an NFS group for example, a CIFS group.

NFSServer.MountReservePortOnly [optional]

Identifies whether all NFS mount requests must be received on a secure port that is reserved for NFS requests only.

- Enable the option if NFS mount requests must be received on a secure port that is reserved for NFS requests (including mount requests). The port number must be less than 1024.
- Disable the option if NFS mount requests can be received on any open port.

NFSServer.AllowNonRootChown [optional]

Identifies whether users who do not have root privileges can change the owner of files that are stored in the Pillar Axiom system.

- Enable the option if non-root users can change ownership on only files that they own.

- Disable the option if only root users can change file ownership.

Note: When quotas are enabled, you may set this flag to automatically transfer files to other users if an individual user exceeds a quota limit.

NFSServer.TCPSupported [optional]

Identifies whether Transmission Control Protocol (TCP) connections are allowed.

- Enable TCP if users can mount filesystems using the Transmission Control Protocol (TCP).
- Disable TCP if users can mount filesystems using only the User Datagram Protocol (UDP).

NFSServer.TCPConnectionsLimit [optional]

Identifies how many concurrent protocol-specific connections are allowed. Enter a value that is large enough to meet your anticipated work load.

- A lower number allows fewer users to access data simultaneously. Once the limit is reached, access to additional users is denied. Set this value to a lower number only if there is a specific reason to restrict the number of NFS TCP connections.
- A higher number allows more users to access data simultaneously. There is no performance penalty associated with setting this value to the maximum (65,000).

NFSCharacterSet [optional]

Identifies which character set is valid. Choose from:

- Standard-ASCII, which is a 7-bit character set that includes 128 printable and control character values (0-127).
- UTF8, which is a Unicode-coded character set that uses an 8-bit transformation format. Defined in RFC 2279.
- ISO8859-1, which is an 8-bit, single-byte character set that is an extension of Standard-ASCII.

EXAMPLES

Run the following command to create a File Server named Company-XYZ-FS1.

```
pdscli request CreateFileServer @input
```

where input.req contains the following name/value pairs:

```
Request=CreateFileServer
Name=Company-XYZ-FS1
Comment=Engineering File Server
PrimaryVirtualInterface.IPAddress=
     172.18.65.128
PrimaryVirtualInterface.IPNetmask=
     255.255.255.0
PrimaryVirtualInterface.VLANTag=1
PrimaryVirtualInterface.FrameSize=1500
PrimaryVirtualInterface.
     NetworkInterfaceIOPort.Slammer=Slammer1
PrimaryVirtualInterface.
     NetworkInterfaceIOPort.ControlUnitNumber=1
PrimaryVirtualInterface.
     NetworkInterfaceIOPort.EthernetPortName=PORT0
DefaultRoute.GatewayIPAddress=172.18.65.1
DNSConfig.DomainName=xyz.eng.corp
DNSConfig.Server1IPAddress=172.18.65.10
DNSConfig.Server2IPAddress=172.18.65.11
DNSConfig.Server3IPAddress=172.18.65.12
NISConfig.DomainName=cms01.xyz.corp
NISConfig.ServerIPAddress=172.18.35.130
NSSSearchOrder.HostLookupService1=NIS
NSSSearchOrder.HostLookupService2=DNS
NSSSearchOrder.PasswordLookupService1=NIS
NSSSearchOrder.PasswordLookupService2=None
AccountMapping.MapIdenticalNames=true
AccountMapping.UseAllDomains=true
CIFSServer.ServerName=CIFS-server
CIFSServer.Comment=CIFS server
CIFSServer.CharacterSet=Standard-ASCII
CIFSServer.OpportunisticLockingEnabled=true
CIFSServer.WINSServer1IPAddress=72.18.40.130
CIFSServer.DomainName=XYZ-XYZ
CIFSServer.UseAnonymousPDCAccess=true
CIFSServer.PDCAccessAccount.DomainName=
     XYZ_CIFS
CIFSServer.PDCAccessAccount.Username=admin
CIFSServer.PDCAccessAccount.Password=passWord
NFSServer.PortNumber=2049
NFSServer.CharacterSet=UTF8
NFSServer.AnonymousUID=-2
NFSServer.AnonymousGID=-2
NFSServer.MountReservePortOnly=true
NFSServer.AllowNonRootChown=false
NFSServer.TCPSupported=true
NFSServer.TCPConnectionsLimit=100
NFSCharacterSet=UTF8
```

See the CreateFileServerResponse.txt file for example response data.

SEE ALSO

CreateRoute on p. 157

CreateVirtualInterface on p. 164

ModifyFileServer on p. 321

DeleteFileServer on p. 175

GetAllFileServers on p. 213

GetFileServerDetails on p. 253

GetFileServerStatistics on p. 254

Ranges for Option Values on p. 57

NAME **CreateFileSystem**

DESCRIPTION Creates a NAS filesystem in a Pillar Axiom storage system. A filesystem is defined as:

> A logical volume that organizes and catalogs files and assigns resources to a given collection of directories and files in a network attached storage (NAS) system. Administrators can assign different Quality of Service (QoS) attributes to each filesystem. A filesystem must be associated with a File Server.

SYNTAX

```
pdscli {request | submit} CreateFileSystem
  FileServer=GUID-or-FQN-of-File-Server
  VolumeGroup=GUID-or-FQN-of-volume-group
  Name=filesystem-name
  Profile={OracleASM | Normal | HighThroughput}
  size=integer-0-through-2147483647
  max=integer-1-through-2147483647
  RelativePriority=
    {Premium | High | Medium | Low | Archive}
  FileSizeBias={None | Large | Medium |Small}
  AccessBias={Sequential | Random | Mixed | None}
  IOBias={Read | Write | Mixed | None}
  StorageClass={MultiLevelCellSolidStateDrives |
    SingleLevelCellSolidStateDrives | FibreChannelHardDiskDrives |
    SataHardDiskDrives}
  Redundancy={Standard | Double}
  CreateDefaultFileSystemSnapshotSchedule=
    {true | false}
  CreateRootDirectoryQuota.EnforceAllLimitsForDirectory=
    {true | false}
  CreateRootDirectoryQuota.SoftLimitInMB=
    {0-for-unlimited | integer}
  CreateRootDirectoryQuota.HardLimitInMB=
    {0-for-unlimited | integer}
  CreateRootDirectoryQuota.MBGracePeriodInDays=
    {0-for-unlimited | integer}
  CreateDefaultFileSystemSnapshotSchedule={true | false}
  EnforceFileRetention={true | false}
  RetentionPolicy.EnforceVolumeRetention={true | false}
  RetentionPolicy.MinimumRetentionPeriod.Units=
    {years | months | days}
  RetentionPolicy.MinimumRetentionPeriod.Amount=
    {0-for-unlimited | integer}
  RetentionPolicy.MaximumRetentionPeriod.Units=
    {years | months | days}
  RetentionPolicy.MaximumRetentionPeriod.Amount=
   {0-for-unlimited | integer}
RetentionPolicy.DefaultRetentionPeriod.Units=
   {years | months | days}
RetentionPolicy.DefaultRetentionPeriod.Amount=
   {0-for-unlimited | integer}
RetentionPolicy.DataIntegrityScanSchedule.StartTime=
   {YYYY-MM-DD-THH:mm:SS.xx+-HH:mm}
CloneFSCapacity={0-for-unlimited | integer}
Slammer=GUID-or-FQN-of-Slammer
ControlUnitNumber={0-for-automatic-assignment | integer}
BackgroundCopyPriority=
   {MaximizeSpeed | MinimizeImpact | SystemChooses}
```

OPTIONS

FileServer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of a File Server from which the filesystem gets protocol-access options.

VolumeGroup

Enter the GUID or FQN of the volume group with which the filesystem is associated.

You are not required to enter a volume group; however, If you do not enter a volume group, the filesystem will be created in the default volume group named "volumes."

Name

Identifies the name that is assigned to a filesystem. Each filesystem name must be unique within its associated File Server and volume group.

You must specify the File Server and the filesystem name, which results in a Fully Qualified File System name of /fileserver/filesystemname.

Profile [optional]

Identifies a performance profile. Choose from:

- Normal, which defines specific performance and redundancy settings for each logical volume. This value is highly recommended and should be used in almost all cases.
- OracleASM, which uses a wide stripe RAID geometry consisting of 1 MB strips to improve the overall performance of logical volumes in an Oracle ASM environment.
- HighThroughput, which stripes the filesystem or LUN across all RAID arrays. SATA Bricks have two RAID groups per Brick, and FC Bricks have one RAID group per Brick. You do not need to set double redundancy to get full array striping.

Important! Use of the HighThroughput profile is for special purposes only. Do not use this profile for ordinary filesystems or LUNs. Please contact the Pillar Customer Services Advanced Solutions Group before using this profile option, which should only be used for special testing where the system configuration is cleared after the test.

size

Identifies the initial capacity, in MB, that is assigned to the object. This value is a soft limit, which means that data can be stored in a logical volume until the maximum capacity is reached.

max

Identifies the maximum capacity, in MB, to which a given object can grow.

RelativePriority

The Relative Priority determines how much of the system resources are devoted to that volume, including the allocation of Slammer CPU cycles and the allocation of specific portions of the disk platters. The higher the Priority, the greater the allocation of CPU time and the faster the media access time.

The Relative Priority option prioritizes the layout of data that is stored in the Pillar Axiom system based on one of the following settings:

- Premium priority is the highest possible performance. This priority uses the outer 20% of the drive platters.

Note: I/O is typically faster when data resides on the outer edge of the drive platters.

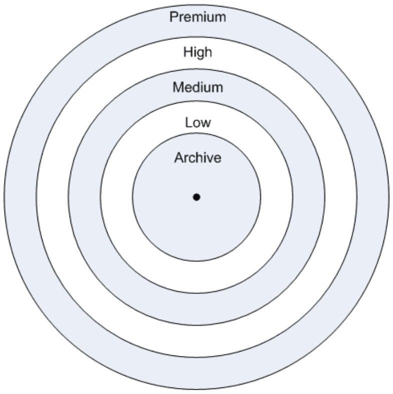
- High priority logical volumes are allocated space in the outer 20%-40% of the drive platters.

Tip: If sufficient Bricks are present, performance of high priority volumes is enhanced by placing those volumes on a greater number of Bricks.

- Medium priority logical volumes are allocated space in the outer 40% to 60% of the drive platters.
- Low priority logical volumes are allocated space in the band that is 60% to 80% from the outer diameter of drive platters.

- Archive (lowest-priority) logical volumes are allocated space in the inner areas of the drive platters.

Figure 1 Example of HDD priority bands



If needed, the Pillar Axiom system allocates space in a higher or lower priority band (within the same Storage Class) than the one that you choose.

FileSizeBias

Identifies the typical size of files that are stored in the specified logical volume. Choose from:

- Small if files are smaller than 20 kilobytes (KB).
- Medium if files are larger than 20KB and smaller than 4 megabytes (MB).
- Large if files are larger than 4MB.

AccessBias

Identifies the typical data access method. Choose from:

Table 13 Data access methods

Access bias	I/O bias	Read ahead in the Slammer	RAID configuration in the Brick
Sequential	Read	Aggressive	RAID 5 Reads large extents of the data into memory.
	Write	Conservative	RAID 5 • Allocates capacity in large, full-stripe (640 K) extents. • Writes data to physical storage more quickly and in full-stripe extents.

Table 13 Data access methods (continued)

Access bias	I/O bias	Read ahead in the Slammer	RAID configuration in the Brick
Mixed and random	Read	None	RAID 5 • Reads small extents into memory. • Allocates capacity in small chunks. • Writes data to physical storage more slowly to combine (or eliminate) write operations.
Random	Write	None	RAID 10

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

IOBias

Identifies the typical read-write ratio. Choose from:

- Read if users or applications read data more often than they write to the data source.
- Write if users or applications write data more often than they read it.

Important! If you choose Random as the access method and Write as the I/O Bias, the system creates the volume with a Distributed RAID geometry. This geometry provides enhanced write performance but uses twice the capacity.

- Mixed if the read-write ratio varies.

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

The system stores all writes of user data and system metadata in mirrored copies of the journal. Although the filesystem or LUN is journaled, for some of the profiles, write caching is disabled because it tends to have very low hit rates.

One copy is maintained in non-volatile memory on one control unit (CU) of a Slammer. The mirror copy is maintained in one of:

- Battery-backed memory of the partner CU on the Slammer (preferred location). Writes to this copy are equivalent to write-back cache.
- An area reserved on disk for the logical volume if the partner CU is unavailable for the write. Writes to this copy are equivalent to write-through cache.

Writes from the journal to permanent disk storage are equivalent to write-through cache. The system flushes user data and the corresponding metadata as a unit to disk.

StorageClass

Identifies the category of physical storage on which the logical volume resides:

- Fibre Channel drives
- SATA drives
- SATA single-level cell solid state drives
- SATA multi-level cell solid state drives

Redundancy

Identifies how many mirror copies of the original data are stored online.

Important! Pillar highly recommends that you consult with a Pillar customer support professional for assistance with sizing your system and creating your volumes (filesystems and LUNs).

Redundancy options include:

- Standard, to store original data only. Data striping over multiple RAID groups maintains full redundancy, even without mirror copies.
- Double, to store original data and one mirror copy, with data striping over multiple RAID groups.

Note: Double Redundancy can only provide true redundancy if your system has enough Bricks to allocate the filesystem or LUN such that no two mirror copies share a RAID group.

SATA Bricks have two RAID groups per Brick, FC Bricks have one RAID group per brick.

If the storage pool is becoming depleted, or a large filesystem or LUN is created, it might be necessary to place the filesystem or LUN on more RAID group fragments.

Depending on the system’s ability to allocate sufficient contiguous storage blocks for the size of the logical volume, use the following number of RAID groups to configure your volumes for best performance:

Table 14 Optimum number of RAID groups for best performance

Priority	SATA standard Redundancy	SATA double redundancy	FS standard redundancy	FS double redundancy
Archive	4	8	2	4
Low	4	8	2	4
Medium	6	12	3	6
High	8	16	4	8
Premium	8	16	4	8

For *performance testing purposes only*, create a filesystem or LUN using standard redundancy and the High Throughput Profile. This is NOT recommended for most applications. Reset your system after you have created a High Throughput LUN or filesystem before you configure normal filesystems or LUNs for applications.

CreateRootDirectoryQuota.EnforceAllLimitsFor Directory [optional]

Identifies whether the quota limits are enforced as well as tracked. All quotas have automatic usage tracking. Enforcement is optional and must be enabled explicitly and separately for each quota.

- Enable this option so that the Pillar Axiom system enforces the capacity limits.
- Disable this option so that the system does not enforce the capacity limits.

CreateRootDirectoryQuota.SoftLimitInMB [optional]

Identifies a soft capacity limit, which means that once the system reaches this value, it sends an administrator alert. If space is available, the system stores data until it reaches the grace period or the hard limit.

Be aware that:

- If a parent directory’s quota limit is smaller than a child directory’s, the system enforces the parent’s limit.
- Users sometimes delete directories from a filesystem's directory structure. If Directory, Directory & User, or Directory & Group quotas are assigned to the deleted directory, the quotas are also deleted.

To define an unlimited value, enter 0 (zero).

CreateRootDirectoryQuota.HardLimitInMB [optional]

Identifies a hard capacity limit, which means that once the system reaches this value, it rejects write requests and does not store data.

To define an unlimited value, enter 0 (zero).

**CreateRootDirectoryQuota.MBGracePeriodIn
Days [optional]**

Identifies the number of days that the system can exceed a soft limit. When the time expires, the soft limit becomes a hard limit.

To define an unlimited value, enter 0 (zero).

CreateDefaultFileSystemSnapshotSchedule [optional]

A snapshot schedule is automatically created unless you disable it.

Allows you to schedule the default filesystem Snap FS. A Snap FS is defined as:

A point-in-time, read-only copy of a filesystem that you intend to snap (split) from the source filesystem for immediate access. A Snap FS retains the QoS parameters from the source filesystem and consumes part of the source filesystem’s storage capacity.

Creating filesystem snapshots is recommended. You can use them to recover accidentally deleted files and for quick file system recovery.

CreateDefaultFileSystemSnapshotSchedule

Creates a replication schedule that creates a Snap FS of a filesystem at regular intervals.

EnforceFileRetention

Forces the retention policy settings, which protect files on a filesystem.

Files written over CIFS connections are protected immediately, not allowing any further modifications.

Files written over NFS connections must be protected explicitly by setting the Immutable field for the file in its corresponding attribute file.

For instructions on how to edit the attribute file, refer to the *Pillar Axiom Administrator's Guide*.

RetentionPolicy.EnforceVolumeRetention

Forces the retention policy settings for a logical volume on a filesystem.

**RetentionPolicy.MinimumRetentionPeriod.
Units**

Identifies the minimum retention period. Valid values are years, months, days.

**RetentionPolicy.MinimumRetentionPeriod.
Amount**

Identifies the minimum retention period. Valid values are 0 to 2147483647.

To define an unlimited value, enter 0 (zero).

**RetentionPolicy.MaximumRetentionPeriod.
Units**

Identifies the maximum retention period. Valid values are years, months, and days.

**RetentionPolicy.MaximumRetentionPeriod.
Amount**

Identifies the minimum retention period. Valid values are 0 to 2147483647 and must be greater than or equal to the minimum retention value.

To define an unlimited value, enter 0 (zero).

RetentionPolicy.DefaultRetentionPeriod.Units

Identifies the maximum retention period. Valid values are years, months, and days.

**RetentionPolicy.DefaultRetentionPeriod.
Amount**

Identifies the minimum retention period. Valid values are 0 to 2147483647 and must be greater than or equal to the minimum retention value.

To define an unlimited value, enter 0 (zero).

RetentionPolicy.DataIntegrityScanSchedule.StartTime

Sets the start time for the data integrity scan, which checks all protected files on the filesystem and validates their data integrity.

CloneFSCapacity

Specifies the maximum amount of repository space to be reserved for Clone FSs from the total system capacity.

⚠ CAUTION We strongly recommend that you allocate sufficient repository capacity to minimize the chances of running out of this space (which could lead to data inconsistency or loss). To set sufficient capacity, use a value equal to the source volume capacity times the number of replicas times the maximum rate of change. For example, for a 100 GB volume that is projected to have 20 active replicas at a time and no more than a 20% rate of change, use a value of 400 GB for the clone repository.

Maximum capacity for Clone FSs is 2147483647.

Slammer [optional]

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer with which the filesystem is associated.

Note: If you do not enter a value, the system will automatically designate a Slammer. If AutomaticallyAssignSlammer is false, the specified value changes the Slammer assignment.

ControlUnitNumber [optional]

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs. The CUs are identified by 0 (the top CU) and 1 (the bottom CU).

Note: If you do not enter a value, the system will automatically designate a CU. If AutomaticallyAssignSlammer is false, the specified value changes the CU assignment.

BackgroundCopyPriority

Identifies the strategy the system should use to control the impact on performance when background tasks need to copy or move data from one location in the storage pool to another:

- MinimizeImpact. Restricts the amount of work performed on a loaded system. This option is intended to have a minimal impact on client I/O throughput at the expense of longer copy times.
- SystemChooses. Balances the background copy with the incoming client I/O. This option is the default.
- MaximizeSpeed. Prioritizes the background copy at the expense of client I/O throughput.

Note: When the system is idle or lightly loaded, the above background task maximizes the amount of work done regardless of the option selected.

The following types of operations are affected by the strategy you select:

- Copy
- Restore
- Quality of Service (QoS) Priority changes
- Storage Class changes

Data transfer operations invoked by the Pillar Axiom MaxRep Replication for NAS utility are not affected by the BackgroundCopyPriority setting.

EXAMPLES

Run the following command to create a filesystem for your engineering organization.

```
pdscli request CreateFileSystem @input
```

where input.req contains the following name/value pairs:

```
Request=CreateFileSystem
  FileServer=/Company-XYZ-FS1
  VolumeGroup=/Eng
  Name=EngAtl
  Profile=Normal
  size=350
  grow=50
  max=500
  RelativePriority=High
  FileSizeBias=Small
  AccessBias=Sequential
  IOBias=Write
  StorageClass=FibreChannelHardDiskDrives
  Redundancy=Double
```

```
CreateRootDirectoryQuota.EnforceAllLimits
  ForDirectory=true
CreateRootDirectoryQuota.SoftLimitInMB=
  2390
CreateRootDirectoryQuota.HardLimitInMB=
  3415
CreateRootDirectoryQuota.
  MBGracePeriodInDays=32
CreateDefaultFileSystemSnapshotSchedule=
  false
RetentionPolicy.EnforceFileRetention
  =true
RetentionPolicy.EnforceVolumeRetention
  =true
RetentionPolicy.MinimumRetentionPeriod.
  Amount=0
RetentionPolicy.MinimumRetentionPeriod.
  Units=Days
RetentionPolicy.MaximumRetentionPeriod.
  Amount=0
RetentionPolicy.MaximumRetentionPeriod.
  Units=Days
RetentionPolicy.DefaultRetentionPeriod.
  Amount=0
RetentionPolicy.DefaultRetentionPeriod.
  Units=Days
```

See the CreateFileSystemResponse.txt file for example response data.

SEE ALSO

ModifyFileSystem on p. 330

DeleteFileSystem on p. 176

GetAllFileSystems on p. 214

GetFileSystemDetails on p. 255

GetFileSystemStatistics on p. 260

GetHistoricalFileSystemStatistics on p. 263

PerformFileSystemCheck on p. 423

CreateFileSystemSnapshotSchedule on p. 124

GetAllFileSystemSnapshots on p. 216

GetAllFileSystemSnapshotSchedules on p. 218

GetFileSystemSnapshotDetails on p. 256

GetFileSystemSnapshotScheduleDetails on p. 258

PerformBackgroundFileSystemDuplication on p. 400

PerformFileSystemDuplication on p. 425

PerformRebuildFileSystemRedundancy on p. 474

PerformRestoreFileSystemFromSnapshot on p. 485

Ranges for Option Values on p. 57

NAME

CreateFileSystemSnapshot

DESCRIPTION

Creates a Snap FS of the specified filesystem. A Snap FS is defined as:

> A point-in-time, read-only copy of a filesystem that you intend to snap (split) from the source filesystem for immediate access. A Snap FS retains the QoS parameters from the source filesystem and consumes part of the source filesystem's storage capacity.
>
> Creating filesystem snapshots is recommended. You can use them to recover accidentally deleted files and for quick file system recovery.

Use this request for frequent, nondisruptive backup and restore operations.

SYNTAX

```
pdscli {request | submit} CreateFileSystemSnapshot
   Name=snap-FS-name
   FS=GUID-or-FQN-of-filesystem
```

OPTIONS

Name

Identifies the name that the Pillar Axiom system assigns to a Snap FS.

FS

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem from which the Snap FS is created.

EXAMPLES

Run the following command to create an immediate (not scheduled) Snap FS of the engineering filesystem that is named EngAtl.

```
pdscli request CreateFileSystemSnapshot @input
```

where input.req contains the following name/value pairs:

```
Request=CreateFileSystemSnapshot
Name=EngAtl-snap
FS=/Company-XYZ-FS1/EngAtl
```

See the CreateFileSystemSnapshotResponse.txt file for example response data.

SEE ALSO

CreateJob on p. 127

- CreateFileSystemSnapshotSchedule on p. 124
- DeleteFileSystemSnapshot on p. 177
- DeleteFileSystemSnapshotSchedule on p. 179
- GetAllFileSystemSnapshots on p. 216
- GetAllFileSystemSnapshotSchedules on p. 218
- GetFileSystemSnapshotDetails on p. 256
- GetFileSystemSnapshotScheduleDetails on p. 258
- ModifyFileSystemSnapshot on p. 340
- PerformFileSystemDuplication on p. 425

NAME **CreateFileSystemSnapshotSchedule**

DESCRIPTION Allows you to automatically create a Snap FS at designated times. A Snap FS is defined as:

A point-in-time, read-only copy of a filesystem that you intend to snap (split) from the source filesystem for immediate access. A Snap FS retains the QoS parameters from the source filesystem and consumes part of the source filesystem's storage capacity.

Creating filesystem snapshots is recommended. You can use them to recover accidentally deleted files and for quick file system recovery.

SYNTAX

```
pdscli {request | submit}
  CreateFileSystemSnapshotSchedule
    Name=base-of-schedule-name
    FS=GUID-or-FQN-of-filesystem
    Freq={Once | Daily | Hourly | Weekly}
    Period=integer
    NumSnaps=integer
    StartTime=YYYY-MM-DDTHH:mm:SS.xx+-HH:mm
```

OPTIONS

Name

Identifies the base name that the Pillar Axiom system assigns to a Snap FS that is created by this schedule. The system appends the schedule type (frequency), and an iteration identifier to the base name of each scheduled Snap FS.

For example, if you define a base name of “Eng” for a schedule that creates hourly Snap FSs of the Engineering filesystem, the first Snap FS name is Eng.hourly.00. When the second iteration of the schedule runs, the Eng.hourly.00 Snap FS is renamed to Eng.hourly.01 and the second iteration becomes Eng.hourly.00. The older Snap FS names continue to increment with each iteration of the schedule and the newest Snap FS is always <base>.<type>.00.

FS

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`

- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem from which the scheduled Snap FSs are created.

Freq

Identifies the frequency at which the Pillar Axiom system creates a Snap FS of the specified filesystem. Choose from:

- once
- daily
- hourly
- weekly

Period

Identifies how often the Pillar Axiom system should perform the scheduled operation. Valid values vary based on the schedule's recurrence interval or frequency. Choose from:

Table 15 Schedule recurrence values

Recurrence interval	Valid values
Hourly	1 through 24, inclusive
Daily	1 through 7, inclusive
Weekly	1 through 4, inclusive

NumSnaps

Identifies the maximum number of scheduled snapshots to create. Valid values are based on the schedule's recurrence interval or frequency.

StartTime

Identifies the date and time at which the Pillar Axiom system starts a scheduled operation. Use the following format for the entry:

YYYY-MM-DDTHH:mm:SS.xx+-HH:mm

where:

- YYYY-MM-DD designates a four-digit year, two-digit month, and two-digit day for the date.
- T is a separator that designates the start of the time portion of the string.
- HH:mm:SS designates hours, minutes, and seconds in values for a 24-hour clock.

- xx designates a fraction of a second, to two decimal places.
- +-HH:mm designates the time zone as an offset from Coordinated Universal Time (UTC) in hours and minutes. Include the + or - prefix as appropriate for your time zone in relation to UTC.

EXAMPLES

Run the following command to create a schedule that makes an hourly Snap FS of the engineering filesystem.

```
pdscli request CreateFileSystemSnapshotSchedule
  @input
```

where input.req contains the following name/value pairs:

```
Request=CreateFileSystemSnapshotSchedule
Name=EngAtl-Hourly
FS=/Company-XYZ-FS1/EngAtl
Freq=Hourly
Period=24
NumSnaps=24
StartTime=2004-01-01T12:10:00.-08:00
```

See the CreateFileSystemSnapshotSchedule Response.txt file for example response data.

SEE ALSO

DeleteFileSystemSnapshot on p. 177

DeleteFileSystemSnapshotSchedule on p. 179

GetAllFileSystemSnapshots on p. 216

GetAllFileSystemSnapshotSchedules on p. 218

GetFileSystemSnapshotDetails on p. 256

GetFileSystemSnapshotScheduleDetails on p. 258

ModifyFileSystemSnapshot on p. 340

PerformFileSystemDuplication on p. 425

PerformBackgroundFileSystemDuplication on p. 400

PerformRestoreFileSystemFromSnapshot on p. 485

PerformRebuildFileSystemRedundancy on p. 474

Ranges for Option Values on p. 57

NAME **CreateJob**

DESCRIPTION Creates a Pillar Axiom job. A *job* is a request that you schedule to run at a later time. Based on the type of scheduled operation that the job is associated with, you must complete different sets of name/value pairs.

SYNTAX

```
pdscli {request | submit} CreateJob
  Name=string
  Schedule=GUID-or-FQN-of-schedule
  Enabled={true | false}
  CreateFileSystemSnapshot.Name=string
  CreateFileSystemSnapshot.FS=
    GUID-or-FQN-of-filesystem
  PerformShutdown
  PerformUpdate.UpdatePackage=
    .tar.gz-file-name
  PerformUpdate.UpdateComponentName[1-8]=
    {"Pilot Software" | "Pilot OS" |
    "Slammer Software" | "Slammer PROM" |
    "Slammer Software AX600" | "Slammer PROM AX600" |
    "Brick Firmware" | Brick FC Firmware}
  PerformUpdateDiskDrive.UpdatePackage=
    GUID-or-FQN-of-Brick
  PerformUpdateDiskDrive.
    UpdateComponentName=
    Name-of-updatable-component
  CreateQuota.AllowFileSystemOffline=
    {true | false}
  CreateQuota.FileSystem=GUID-or-FQN-of-
    filesystem
  CreateQuota.Path=path-to-directory
  CreateQuota.RestrictUsers={AllAggregate |
    SingleUser | Group}
  CreateQuota.EnforceAllLimitsForDirectory=
    {true-if-AllAggregate | false-otherwise}
  CreateQuota.Username=string-if-select-
    SingleUser
  CreateQuota.Groupname=string-if-select-
    Group
  CreateQuota.SoftLimitInMB={0-for-unlimited
    | integer}
  CreateQuota.HardLimitInMB={0-for-unlimited
    | integer}
  CreateQuota.MBGracePeriodInDays=
    {0-for-unlimited | integer}
  DeleteQuota.ID=GUID-or-FQN-of-filesystem
  DeleteQuota.AllowFileSystemOffline=
    {true | false}
  PerformUpdateDiskDrive.UpdatePackage=
    GUID-or-FQN-of-filesystem
  PerformUpdate.PerformUpgradeDisruptively=
    {true | false}
  PerformVerifyDataConsistency.Brick=
    GUID-or-FQN-of-Brick
  PerformVerifyDataConsistency.Priority=
    {High | Low}
  PerformVerifyStorageRedundancy.FileSystem=
    GUID-or-FQN-of-filesystem
  PerformVerifySpareDrive.Brick=
    GUID-or-FQN-of-Brick
  PerformVerifySystemConfiguration
```

OPTIONS

Name

Identifies the name of the scheduled job.

Schedule

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the schedule with which this job is associated. A null indicates that the job is not scheduled.

Enabled [optional]

Identifies whether the scheduled job is enabled.

- Enable this option to perform the job at the scheduled time.
- Disable this option to hold the scheduled job until you enable it at a later time.

CreateFileSystemSnapshot.Name [optional]

Identifies the name that is assigned to an administrator-managed Snap FS.

CreateFileSystemSnapshot.FS [optional]

Enter the GUID or FQN of the filesystem from which the data replica is created.

PerformShutdown [optional]

Schedules a system shutdown.

PerformUpdate.UpdatePackage [optional]

Identifies the name of an update file.

- The staging process uploads the file to the Pilot and expands the compressed .tar.gz file.
- The validation process checks the compatibility of the firmware and software in the update file against previously installed firmware and software.
- The update process installs the uncompressed package on the Pillar Axiom storage system.

Select a single update file for the given operation.

PerformUpdate.UpdateComponentName[1-5] [optional]

Identifies the name of an updatable component. Choose between one and six of the following options.

Table 16 Updatable components

Name	Description
“Pilot Software”	Software that runs on the Pilot, such as the GUI interface and web server, online help, SNMP, and NDMP
“Pilot OS”	Operating system for the Pilot
“Slammer Software”	NAS or SAN software that runs on Pillar Axiom 300 and 500 Slammers
“Slammer Software AX600”	NAS or SAN software that runs on Pillar Axiom 600 Slammers
“Slammer PROM”	Programmable ROM (PROM), that includes BIOS and netboot code, for Pillar Axiom 300 and 500 NAS or SAN Slammers
“Slammer PROM AX600”	Programmable ROM (PROM), that includes BIOS and netboot code, for Pillar Axiom 600 Slammers
“Brick Firmware”	RAID firmware for SATA Bricks
“Brick FC Firmware”	FC firmware for Bricks
“Brick SATA2 Firmware”	RAID firmware for version 2 SATA Bricks

PerformUpdateDiskDrive.UpdatePackage [optional]

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick to update.

PerformUpdateDiskDrive.
UpdateComponentName [optional]

Identifies the name of the updatable Brick.

CreateQuota.AllowFileSystemOffline [optional]

Identifies whether the Pillar Axiom system can take the filesystem offline to perform quota operations.

- Enable this option if the Pillar Axiom system can take the filesystem offline to create or remove a directory-level quota when the specified directory contains files and subdirectories.
- Disable this option if the system cannot take the filesystem offline to perform the quota operation when the specified directory contains files and subdirectories. The filesystem remains online, and the directory-level quota is not created or removed.

Whichever setting you choose, the filesystem remains online and the quota operation is performed when the specified directory is empty.

CreateQuota.FileSystem [optional]

Enter the GUID or FQN of the filesystem for which the quota is to be created.

CreateQuota.Path [optional]

Enter a full path, which starts with a forward slash (/), to the directory that is covered by this quota.

CreateQuota.RestrictUsers [optional]

Identifies who the quota covers. Choose from:

- AllAggregate if the quota covers all users who store data in the specified directory. Leave the Username and Groupname fields blank.
- SingleUser if the quota covers one user. Identify which user the quota covers in the Username field.
- Group if the quota covers a group of users. Identify which group the quota covers in the Groupname field.

CreateQuota.EnforceAllLimitsForDirectory [optional]

Identifies whether the system enforces and tracks quota limits. All quotas have automatic usage tracking. You can optionally enable enforcement for each quota.

- If RestrictUsers=AllAggregate, enable this option so that the Pillar Axiom system enforces all quota limits in the specified directory.
- If RestrictUsers=SingleUser or Group, disable this option. The system tracks and enforces the user’s or group’s quota limits in the filesystem.

CreateQuota.Username [optional]

Identifies a user that the quota covers. Enter a user name, user ID (UID), CIFS account name, or NFS account name. Field types support UTFString.

CreateQuota.Groupname [optional]

Identifies a group of users that the quota covers. Enter a group name or group ID (GID).

CreateQuota.SoftLimitInMB [optional]

Identifies a soft capacity limit, which means that once the system reaches this value, it sends an administrator alert. If space is available, the system stores data until it reaches the grace period or the hard limit.

Be aware that:

- If a parent directory’s quota limit is smaller than a child directory’s, the system enforces the parent’s limit.
- Users sometimes delete directories from a filesystem's directory structure. If Directory, Directory & User, or Directory & Group quotas are assigned to the deleted directory, the quotas are also deleted.

To define an unlimited value, enter 0 (zero).

CreateQuota.HardLimitInMB [optional]

Identifies a hard capacity limit, which means that once the system reaches this value, it rejects write requests and does not store data.

To define an unlimited value, enter 0 (zero).

CreateQuota.MBGracePeriodInDays [optional]

Identifies the number of days that the system can exceed a soft limit. When the time expires, the soft limit becomes a hard limit.

To define an unlimited value, enter 0 (zero).

DeleteQuota.ID [optional]

Enter the GUID or FQN of the filesystem from which the quota is to be removed.

DeleteQuota.AllowFileSystemOffline [optional]

Identifies whether the Pillar Axiom system can take the filesystem offline to perform quota operations.

- Enable this option if the Pillar Axiom system can take the filesystem offline to create or remove a directory-level quota when the specified directory contains files and subdirectories.
- Disable this option if the system cannot take the filesystem offline to perform the quota operation when the specified directory contains files and subdirectories. The filesystem remains online, and the directory-level quota is not created or removed.

Whichever setting you choose, the filesystem remains online and the quota operation is performed when the specified directory is empty.

PerformUpdateDiskDrive.UpdatePackage

Identifies the name of the update package for the disk drive.

PerformUpdate.PerformUpgradeDisruptively

Upgrades are typically performed non-disruptively. Enable this option only when advised by Pillar customer services, unless it is necessary to set it for an upgrade that defines itself as disruptive.

PerformVerifyDataConsistency.Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick to verify.

PerformVerifyDataConsistency.Priority [optional]

Identifies the priority to assign to the verification check. Choose from:

- High to permit the verification check to affect performance by up to 30%.
- Low to permit the verification check to affect performance by up to 10%.

PerformVerifyStorageRedundancy.FileSystem [optional]

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem to verify.

PerformVerifySpareDrive.Brick [optional]

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick that contains the spare disk drive to verify.

PerformVerifySystemConfiguration [optional]

Verifies the system configuration.

EXAMPLES

Run the following command to schedule an update of the Slammer software.

```
pdscli request CreateJob @input
```

where input.req contains the following name/value pairs:

```
Request=CreateJob
Name=upgrade-software
Schedule=IDA0479DC2-38EF-9828-8902-
     E0FC03743270
Enabled=true
PerformUpdate.StagedPackage=
  mgmt_upgrade0003.tar.gz
PerformUpdate.UpdateComponentName[1]=
   "Slammer Software"
PerformUpdate.PerformUpgradeDisruptively=false
PerformVerifySystemConfiguration
```

See the CreateJobResponse.txt file for example response data.

SEE ALSO

CreateQuota on p. 154

DeleteQuota on p. 186

GetAllUpdatePackages on p. 237

GetUpdatePackageDetails on p. 305

PerformUpdate on p. 507

PerformUpdate on p. 507

PerformUpdate on p. 507

PerformUpdate on p. 507

PerformUpdateValidation on p. 513

PerformVerifyDataConsistency on p. 515

PerformVerifyStorageRedundancy on p. 516

ModifyJob on p. 345

DeleteJob on p. 181

Ranges for Option Values on p. 57

NAME

CreateLUN

DESCRIPTION

Creates a LUN in a Pillar Axiom storage system to efficiently partition (stripe) storage across multiple Bricks based on storage or performance priorities. This is the LUN that is exposed to the client as a SAN LUN (SLUN)

A LUN is defined as:

> A logical volume within a storage area network (SAN). Administrators assign storage resources and Quality of Service (QoS) attributes to each logical unit (LUN).

SYNTAX

```
pdscli {request | submit} CreateLUN
  EnableiSCSIAccess={true | false}
  EnableFibreChannelAccess={true | false}
  VolumeGroup=GUID-or-FQN-of-volume-group
  Name=LUN-name
  Slammer=GUID-or-FQN-of-Slammer
  ControlUnitNumber={0 | 1}
  LUNNumber=integer-0-through-255
  Mapped={true | false}
  Profile={OracleASM | Normal | HighThroughput}
  Capacity=integer-0-through-2147483647
  Max=integer-1-through-2147483647
  RelativePriority={Archive | Low | Medium | High | Premium}
  AccessBias={None | Mixed | Random | Sequential}
  IOBias={Mixed | Write | Read | Write | None}
  StorageClass={MultiLevelCellSolidStateDrives |
    SingleLevelCellSolidStateDrives | FibreChannelHardDiskDrives |
    SataHardDiskDrives}
  Redundancy={Standard | Double}
  SuppressConservativeMode{true | false}
  SnapLUNStorage.MaximumCapacity=integer-0-through-2147483647
  CloneStorage.MaximumCapacity=integer-0-through-2147483647
  BackgroundCopyPriority=
    {MaximizeSpeed | MinimizeImpact | SystemChooses}
```

OPTIONS

EnableiSCSIAccess

Identifies that the iSCSI protocol specification is used when accessing the newly created SAN LUN.

Note: You must specify iSCSI, Fibre Channel, or both protocols. If you do not specify a protocol, the creation task will fail.

EnableFibreChannelAccess

Identifies that the Fibre Channel (FC) protocol specification is used when accessing the newly created SAN LUN.

Note: You must specify iSCSI, Fibre Channel, or both protocols. If you do not specify a protocol, the creation task will fail.

VolumeGroup

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the volume group with which the LUN is associated.

Name

Identifies the name that is assigned to a LUN for administrative purposes. LUN names must be unique across the Pillar Axiom system, not just within its associated volume group.

Name is a required parameter only if you want to change the FQN LUN's user-visible name.

Slammer [optional]

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer with which the LUN is associated.

Note: If you do not enter a value, the system will automatically designate a Slammer.

ControlUnitNumber [optional]

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs. The CUs are identified by 0 (the top CU) and 1 (the bottom CU).

Note: If you do not enter a value, the system will automatically designate a CU.

LUNNumber [optional]

Identifies the unique number that is assigned to a LUN and can be accessed by all SAN hosts if the LUN is not mapped. If the LUN is mapped, the LUN number must be unique to the mapped SAN host.

Note: If you intend to map this LUN to a specified host, do not include the LUNNumber option.

Mapped [optional]

Identifies whether all hosts or only selected hosts may access this LUN.

- Mapped allows only designated hosts to access this LUN.
- Unmapped allows all hosts to access this LUN using the LUN Number.

Note: If you choose Mapped, do not assign a LUNNumber.

Profile [optional]

Identifies a performance profile. Choose from:

- Normal, which defines specific performance and redundancy settings for each logical volume. This value is highly recommended and should be used in almost all cases.
- OracleASM, which uses a wide stripe RAID geometry consisting of 1 MB strips to improve the overall performance of logical volumes in an Oracle ASM environment.
- HighThroughput, which stripes the filesystem or LUN across all RAID arrays. SATA Bricks have two RAID groups per Brick, and FC Bricks have one RAID group per Brick. You do not need to set double redundancy to get full array striping.

Important! Use of the HighThroughput profile is for special purposes only. Do not use this profile for ordinary filesystems or LUNs. Please contact the Pillar Customer Services Advanced Solutions Group before using this profile option, which should only be used for special testing where the system configuration is cleared after the test.

Capacity

Identifies the capacity that is assigned to this LUN.

Max [optional]

Identifies the maximum capacity limit that is assigned to the object.

RelativePriority

The Relative Priority determines how much of the system resources are devoted to that volume, including the allocation of Slammer CPU cycles and the allocation of specific portions of the disk platters. The higher the Priority, the greater the allocation of CPU time and the faster the media access time.

The Relative Priority option prioritizes the layout of data that is stored in the Pillar Axiom system based on one of the following settings:

- Premium priority is the highest possible performance. This priority uses the outer 20% of the drive platters.

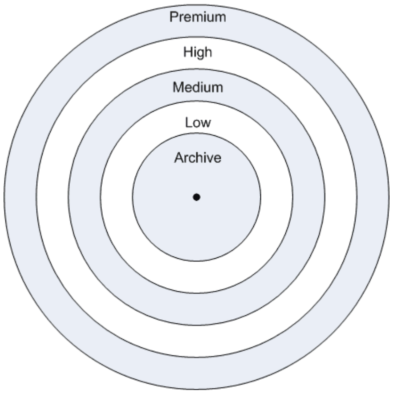
Note: I/O is typically faster when data resides on the outer edge of the drive platters.

- High priority logical volumes are allocated space in the outer 20%-40% of the drive platters.

Tip: If sufficient Bricks are present, performance of high priority volumes is enhanced by placing those volumes on a greater number of Bricks.

- Medium priority logical volumes are allocated space in the outer 40% to 60% of the drive platters.
- Low priority logical volumes are allocated space in the band that is 60% to 80% from the outer diameter of drive platters.
- Archive (lowest-priority) logical volumes are allocated space in the inner areas of the drive platters.

Figure 2 Example of HDD priority bands



If needed, the Pillar Axiom system allocates space in a higher or lower priority band (within the same Storage Class) than the one that you choose.

AccessBias

Identifies the typical data access method. Choose from:

Table 17 Data access methods

Access bias	I/O bias	Read ahead in the Slammer	RAID configuration in the Brick
Sequential	Read	Aggressive	RAID 5 Reads large extents of the data into memory.
	Write	Conservative	RAID 5 • Allocates capacity in large, full-stripe (640 K) extents. • Writes data to physical storage more quickly and in full-stripe extents.
Mixed and random	Read	None	RAID 5 • Reads small extents into memory. • Allocates capacity in small chunks. • Writes data to physical storage more slowly to combine (or eliminate) write operations.
Random	Write	None	RAID 10

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

IOBias

Identifies the typical read-write ratio. Choose from:

- Read if users or applications read data more often than they write to the data source.
- Write if users or applications write data more often than they read it.

Important! If you choose Random as the access method and Write as the I/O Bias, the system creates the volume with a Distributed RAID geometry. This geometry provides enhanced write performance but uses twice the capacity.

- Mixed if the read-write ratio varies.

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

The system stores all writes of user data and system metadata in mirrored copies of the journal. Although the filesystem or LUN is journaled, for some of the profiles, write caching is disabled because it tends to have very low hit rates.

One copy is maintained in non-volatile memory on one control unit (CU) of a Slammer. The mirror copy is maintained in one of:

- Battery-backed memory of the partner CU on the Slammer (preferred location). Writes to this copy are equivalent to write-back cache.
- An area reserved on disk for the logical volume if the partner CU is unavailable for the write. Writes to this copy are equivalent to write-through cache.

Writes from the journal to permanent disk storage are equivalent to write-through cache. The system flushes user data and the corresponding metadata as a unit to disk.

StorageClass

Identifies the category of physical storage on which the logical volume resides:

- Fibre Channel drives
- SATA drives
- SATA single-level cell solid state drives
- SATA multi-level cell solid state drives

Redundancy

Identifies how many mirror copies of the original data are stored online.

Important! Pillar highly recommends that you consult with a Pillar customer support professional for assistance with sizing your system and creating your volumes (filesystems and LUNs).

Redundancy options include:

- Standard, to store original data only. Data striping over multiple RAID groups maintains full redundancy, even without mirror copies.
- Double, to store original data and one mirror copy, with data striping over multiple RAID groups.

Note: Double Redundancy can only provide true redundancy if your system has enough Bricks to allocate the filesystem or LUN such that no two mirror copies share a RAID group.

SATA Bricks have two RAID groups per Brick, FC Bricks have one RAID group per brick.

If the storage pool is becoming depleted, or a large filesystem or LUN is created, it might be necessary to place the filesystem or LUN on more RAID group fragments.

Depending on the system's ability to allocate sufficient contiguous storage blocks for the size of the logical volume, use the following number of RAID groups to configure your volumes for best performance:

Table 18 Optimum number of RAID groups for best performance

Priority	SATA standard Redundancy	SATA double redundancy	FS standard redundancy	FS double redundancy
Archive	4	8	2	4
Low	4	8	2	4
Medium	6	12	3	6
High	8	16	4	8
Premium	8	16	4	8

For *performance testing purposes only*, create a filesystem or LUN using standard redundancy and the High Throughput Profile. This is NOT recommended for most applications. Reset your system after you have created a High Throughput LUN or filesystem before you configure normal filesystems or LUNs for applications.

SuppressConservativeMode

Identifies that the LUN continues to use write-back cache even if the control unit (CU) is in a degraded mode.

Important! Before setting SuppressConservativeMode to TRUE, consult with the Pillar World Wide Customer Support Center to discuss the implications of this mode. When operating in this mode, data loss will occur when operating with a failed control unit when a power failure occurs.

SnapLUNStorage.MaximumCapacity

Identifies additional capacity used for snapshot storage that is assigned to this LUN. As a guideline, create Clone LUN space equal to or greater than the original LUN (100 to 150%). The Clone LUN does not consume space from the original LUN.

Important! If the Clone LUN reaches the maximum capacity limit, the system will notify you and automatically delete any Clone LUN that has exceeded the limit.

CloneStorage.MaximumCapacity

Identifies additional capacity used for snapshot storage that is assigned to this LUN. As a guideline, create Clone LUN space equal to or greater than the original LUN (100 to 150%). The Clone LUN does not consume space from the original LUN.

Important! If the Clone LUN reaches the maximum capacity limit, the system will notify you and automatically delete any Clone LUN that has exceeded the limit.

Note: Both SnapLUNStorage.MaximumCapacity and CloneStorage.MaximumCapacity parameters perform the same function. CloneStorage.MaximumCapacity simply reflects the new naming conventions.

BackgroundCopyPriority

Identifies the strategy the system should use to control the impact on performance when background tasks need to copy or move data from one location in the storage pool to another:

- MinimizeImpact. Restricts the amount of work performed on a loaded system. This option is intended to have a minimal impact on client I/O throughput at the expense of longer copy times.
- SystemChooses. Balances the background copy with the incoming client I/O. This option is the default.
- MaximizeSpeed. Prioritizes the background copy at the expense of client I/O throughput.

Note: When the system is idle or lightly loaded, the above background task maximizes the amount of work done regardless of the option selected.

The following types of operations are affected by the strategy you select:

- Copy
- Restore
- Quality of Service (QoS) Priority changes
- Storage Class changes

Data transfer operations invoked by the Pillar Axiom MaxRep Replication for NAS utility are not affected by the BackgroundCopyPriority setting.

EXAMPLES

Run the following command to create a LUN for the Engineering organization.

```
pdscli request CreateLUN @input
```

```
where input.req contains the following name/value pairs:
Request=CreateLUN
EnableiSCSIAccess=true
EnableFCAccess=false
VolumeGroup=/Eng
Name=EngLUN
LUNNumber=0
Mapped=false
Profile=Normal
Capacity=500
RelativePriority=Archive
AccessBias=Random
IOBias=Write
StorageClass=FibreChannelHardDiskDrives
BackgroundCopyPriority=MinimizeImpact
Redundancy=Standard
```

See the CreateLUNResponse.txt file for example response data.

SEE ALSO

ModifyLUN on p. 351

DeleteLUN on p. 182

GetAllAvailableLUNNumbers on p. 202

GetAllLUNs on p. 221

GetLUNDetails on p. 269

GetLUNStatistics on p. 272

GetHostMappingConfigDetails on p. 265

GetLUNMappingConfigDetails on p. 270

PerformConfigureLUNMapping on p. 412

PerformRebuildLUNRedundancy on p. 475

ModifySnapLUN on p. 381

DeleteSnapLUN on p. 191

GetAllSnapLUNHierarchies on p. 232

GetAllSnapLUNs on p. 233

GetSnapLUNDetails on p. 297

PerformLUNDuplication on p. 447

PerformBackgroundLUNDuplication on p. 402

NAME **CreateNFSExport**

DESCRIPTION Creates a Network File system (NFS) export for a filesystem. An export is a filesystem mount point for NFS clients (users).

SYNTAX

```
pdscli {request | submit} CreateNFSExport
   FileSystem=GUID-or-FQN-of-filesystem
   RootMapping=integer
   path=/path/to/directory
   CreateNFSHostEntry[1-65535].
      HostEntryType=
      {AllHosts | SingleHost | Subnet | NISNetGroup}
   CreateNFSHostEntry[1-65535].
      HostSpecification=host-definition
   CreateNFSHostEntry[1-65535].
      AllowRootAccess={true | false}
   CreateNFSHostEntry[1-65535].
      WriteSupported={true | false}
   CreateNFSHostEntry[1-65535].
      AuthenticationMode=
      {System | None}
   CreateNFSHostEntry[1-65535].
      OrderNumber=integer
```

OPTIONS

FileSystem

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem with which the NFS export is associated. For example, to create an export for the filesystem `test` on the File Server `vault`, use
`/vault/test` as the FQN.

RootMapping

Identifies the user ID (UID) for anonymous users.

- Set the UID to zero (0) if you want anonymous-user access to be identified as the root user. This is a very insecure setting. Setting anonymous to root allows all users full control.

- Set the UID to the "nobody" value (often -2, but not always) if you want anonymous-user access to be identified as the user "nobody." In most instances, this is the recommended setting.
- Set the UID to any other user ID that you want to be identified as anonymous-user access.

path

Enter a full path to the NFS export. To create an export for the filesystem, the path must start with a forward slash (/). Each entry must export a different path name. To assign different permissions to different hosts on a specific export, create multiple host entries for the same export.

For example, if you want to create an export for the directory users in the filesystem `test,` you must first export the filesystem test to a client with root privileges. This export will use a forward slash (/) to export the entire directory.

From the client with root privileges, do the following:

1. Mount the filesystem test to a suitable mount point on the client (for example, /mnt).
2. Create the directory users from this client as /mnt/users and set the permission on users from the client.
3. Export the directory users to other clients, as required.

This export will be for /users, not /test/users.

**CreateNFSHostEntry[1-65535].
HostEntryType**

Identifies whether an NFS host specification represents only one, or more than one, NFS clients, or hosts. Choose from:

- AllHosts
- SingleHost
- Subnet
- NISNetgroup

**CreateNFSHostEntry[1-65535].
HostSpecification**

Identifies the NFS hosts, or clients, that can mount the NFS export. Valid values are based on your host type specification. Choose from:

- Asterisk (*), to export to all NFS hosts (everyone).
- Single IP address in dotted-decimal format.
- Host name. The Pillar Axiom system resolves the host name through the DNS or NIS servers in the order that you specify for NSS Search Settings.
- NIS netgroup name with an @ prefix, if you define NIS in the NSS Search Settings for Hosts and specify an NIS server that contains the netgroup.

- Subnet address with both the subnet and the mask in dotted-decimal format. For example, to define subnet 172.20.129.0/24, enter:

```
172.20.129.0/255.255.255.0
```

CreateNFSHostEntry[1-65535].AllowRootAccess

Identifies whether users who are logged in to a client workstation as root are root users on the export.

- Enable this option so that root users on clients are root users. This is comparable to no_root_squash in Linux root-squashing options.
- Disable this option so that root users on clients are identified as the user nobody. This is comparable to the root_squash option.

CreateNFSHostEntry[1-65535].WriteSupported

Identifies whether the export is made available to users in read-only or read-write mode.

- Enable this option so that the export operates in read-only mode.
- Disable this option so that the export operates in read-write mode.

CreateNFSHostEntry[1-65535].AuthenticationMode

Identifies how users are authenticated.

- System (AUTH_SYS) means that user IDs (UIDs), group IDs (GIDs), and supplemental group IDs are passed with each NFS request. These are authenticated by the server against uploaded group and password NIS-compatible files or from a configured NIS server. The server trusts the client system for authenticating and setting the correct UID and GID.
- None (AUTH_NONE) means that users are mapped to the anonymous user “nobody.” UIDs and GIDs are not passed with the NFS requests. As a result, all files are directories are owned by one user and group.

We recommend that you set this option to System.

CreateNFSHostEntry[1-65535].OrderNumber

Identifies the authentication order for this host relative to other hosts on the export. For this zero-based index, 0 identifies the first host that is authenticated.

We recommend that you set this option to 2.

EXAMPLES

Run the following command to create an export with a host entry of “AllHosts” for the engineering organization’s filesystem.

```
pdscli request CreateNFSExport @input
```

where input.req contains the following name/value pairs:

```
Request=CreateNFSExport
FileSystem=/Company-XYZ-FS1/EngAtl
RootMapping=-2
```

```
path=/src
CreateNFSHostEntry[1].HostEntryType=AllHosts
CreateNFSHostEntry[1].HostSpecification=*
CreateNFSHostEntry[1].AllowRootAccess=false
CreateNFSHostEntry[1].WriteSupported=true
CreateNFSHostEntry[1].AuthenticationMode=
  System
CreateNFSHostEntry[1].OrderNumber=0
```

See the CreateNFSExportResponse.txt file for example response data.

SEE ALSO

CreateNFSHostEntry on p. 148

DeleteNFSExport on p. 183

DeleteNFSHostEntry on p. 184

GetAllNFSExports on p. 222

GetNFSExportDetails on p. 276

ModifyNFSHostEntry on p. 371

Ranges for Option Values on p. 57

NAME **CreateNFSHostEntry**

DESCRIPTION Creates a host entry for an NFS export. A host entry specifies a host or set of hosts that are permitted to mount the export. You can configure each export with multiple host entries.

SYNTAX

```
pdscli {request | submit} CreateNFSHostEntry
  NFSExport=GUID-or-FQN-of-NFS-export
  HostEntryType={AllHosts | SingleHost |
    Subnet | NISNetGroup}
  HostSpecification=host-definition
  AllowRootAccess={true | false}
  WriteSupported={true | false}
  AuthenticationMode={System | None |
    KerberosV5 | Unknown}
  OrderNumber=integer
```

OPTIONS

NFSExport

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the NFS export with which the host entry is associated.

HostEntryType [optional]

Identifies whether an NFS host specification represents only one, or more than one, NFS clients, or hosts. Choose from:

- AllHosts
- SingleHost
- Subnet
- NISNetgroup

HostSpecification

Identifies the NFS hosts, or clients, that can mount the NFS export. Valid values are based on your host type specification. Choose from:

- Asterisk (*), to export to all NFS hosts (everyone).
- Single IP address in dotted-decimal format.
- Host name. The Pillar Axiom system resolves the host name through the DNS or NIS servers in the order that you specify for NSS Search Settings.

- NIS netgroup name with an @ prefix, if you define NIS in the NSS Search Settings for Hosts and specify an NIS server that contains the netgroup.
- Subnet address with both the subnet and the mask in dotted-decimal format. For example, to define subnet 172.20.129.0/24, enter:

```
172.20.129.0/255.255.255.0
```

AllowRootAccess

Identifies whether users who are logged in to a client workstation as root are root users on the export.

- Enable this option so that root users on clients are root users. This is comparable to no_root_squash in Linux root-squashing options.
- Disable this option so that root users on clients are identified as the user nobody. This is comparable to the root_squash option.

WriteSupported

Identifies whether the export is made available to users in read-only or read-write mode.

- Enable this option so that the export operates in read-only mode.
- Disable this option so that the export operates in read-write mode.

AuthenticationMode

Identifies how users are authenticated.

- System (AUTH_SYS) means that user IDs (UIDs), group IDs (GIDs), and supplemental group IDs are passed with each NFS request, without being authenticated by the File Server.
- None (AUTH_NONE) means that users are mapped to the anonymous user “nobody.” UIDs and GIDs are not passed with the NFS requests.

We recommend that you set this option to System.

OrderNumber

Identifies the authentication order for this host relative to other hosts on the export. For this zero-based index, 0 identifies the first host that is authenticated.

We recommend that you set this option to 2.

EXAMPLES

Run the following command to create a host specification for an NFS export.

```
pdscli request CreateNFSHostEntry @input
```

where input.req contains the following name/value pairs:

```
Request=CreateNFSHostEntry
NFSExport=IDA0479DC2-38EF-9828-8902-
  E0FC03743270
HostEntryType=AllHosts
HostSpecification=*
AllowRootAccess=false
```

```
WriteSupported=true
AuthenticationMode=System
OrderNumber=0
```

See the CreateNFSHostEntryResponse.txt file for example response data.

SEE ALSO

ModifyNFSHostEntry on p. 371

DeleteNFSHostEntry on p. 184

Ranges for Option Values on p. 57

NAME

CreateNotificationConnection

DESCRIPTION

Creates a new connection by which client messages are distributed. The Pillar Axiom storage system sends notifications when an event is generated.

SYNTAX

```
pdscli {request | submit}
   CreateNotificationConnection
      Client={SNMP | Email}
```

OPTIONS

Client

Identifies the notification client. Choose from:

- SNMP, if the client is a Simple Network Management Protocol (SNMP) application.
- Email, if the client is an electronic mail (email) server that sends the notifications to designated email addresses.

EXAMPLES

Run the following command to create an SNMP notification connection.

```
pdscli request CreateNotificationConnection @input
```

where input.req contains the following name/value pairs:

```
Request=CreateNotificationConnection
Client=SNMP
```

See the CreateNotificationConnectionResponse.txt file for example response data.

SEE ALSO

CreateNotificationDefinition on p. 152

ModifyNotificationDefinition on p. 373

Ranges for Option Values on p. 57

NAME **CreateNotificationDefinition**

DESCRIPTION Creates a new client message. The Pillar Axiom storage system sends notifications to the specified client when an event of the specified type is generated.

SYNTAX

```
pdscli {request | submit}
   CreateNotificationDefinition
      Name=notification-name
      Description=string
      Client={SNMP | Email}
      EchoDefinitionInNotification={
         true | false}
      SeverityFilter.Severity[1-10]={Critical
         | Error | Warning | Informational}
      EmailAddress[1-4]=username@domain.ext
```

OPTIONS

Name

Identifies the name of a notification that the Pillar Axiom system generates when an event or alert is triggered.

Description [optional]

Describes the notification in a way that is meaningful to the recipients.

Client

Identifies the notification client. Choose from:

- SNMP, if the client is a Simple Network Management Protocol (SNMP) application.
- Email, if the client is an electronic mail (email) server that sends the notifications to designated email addresses.

EchoDefinitionInNotification

Identifies whether the notification definition is included with the event details in the message that the Pillar Axiom system sends to designated recipients.

SeverityFilter.Severity[1-10]

Identifies a category of event severities. Choose from:

- Critical, which means that access to data is compromised (for example, two disks in a RAID array have failed) or data loss has occurred.
- Error, which means that administrator action is required to prevent a “hard” error (for example, a single disk in a RAID array has failed) from becoming a critical event.
- Warning, which means that administrator action is required to prevent a “soft” error from becoming an error or critical event.
- Informational, which means that a configuration change has been detected or another non-error event has occurred.

EmailAddress[1-4]

Identifies the recipient's email address. The email server to which the Pillar Axiom system sends alerts must be able to send messages to this email address.

If you specify Client=Email, enter up to four email addresses. Email addresses are not required if the Client=SNMP.

EXAMPLES

Run the following command to create an email notification for critical events.

```
pdscli request CreateNotificationDefinition @input
```

where input.req contains the following name/value pairs:

```
Request=CreateNotificationDefinition
Name=Critical
Description=Notification for critical events
Client=Email
EchoDefinitionInNotification=false
SeverityFilter.Severity[1]=Critical
EmailAddress1=xavier@xyz.com
EmailAddress2=yvette@xyz.com
```

See the CreateNotificationDefinitionResponse.txt file for example response data.

SEE ALSO

ModifyNotificationDefinition on p. 373

DeleteNotificationDefinition on p. 185

GetAllNotificationDefinitions on p. 223

GetNotificationDefinitionDetails on p. 278

Ranges for Option Values on p. 57

NAME **CreateQuota**

DESCRIPTION Creates a quota on the specified filesystem. Quotas are capacity limits that the Pillar Axiom storage system tracks. Quota enforcement is optional.

SYNTAX

```
pdscli {request | submit} CreateQuota
  AllowFileSystemOffline={true | false}
  FileSystem=GUID-or-FQN-of-filesystem
  Path=/path/to/directory
  RestrictUsers={AllAggregate | SingleUser |
    Group}
  EnforceAllLimitsForDirectory=
    {true-if-AllAggregate | false-otherwise}
  Username=string-if-select-SingleUser
  Groupname=string-if-select-Group
  SoftLimitInMB={0-for-unlimited | integer}
  HardLimitInMB={0-for-unlimited | integer}
  MBGracePeriodInDays={0-for-unlimited |
    integer}
```

OPTIONS

AllowFileSystemOffline

Identifies whether the Pillar Axiom system can take the filesystem offline to perform quota operations.

- Enable this option if the Pillar Axiom system can take the filesystem offline to create or remove a directory-level quota when the specified directory contains files and subdirectories.
- Disable this option if the system cannot take the filesystem offline to perform the quota operation when the specified directory contains files and subdirectories. The filesystem remains online, and the directory-level quota is not created or removed.

Whichever setting you choose, the filesystem remains online and the quota operation is performed when the specified directory is empty.

FileSystem

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem with which the quota is associated.

Path

Enter a full path, which starts with a forward slash (/), to the directory that is covered by this quota.

RestrictUsers

Identifies who the quota covers. Choose from:

- AllAggregate if the quota covers all users who store data in the specified directory. Leave the Username and Groupname fields blank.
- SingleUser if the quota covers one user. Identify which user the quota covers in the Username field.
- Group if the quota covers a group of users. Identify which group the quota covers in the Groupname field.

EnforceAllLimitsForDirectory [optional]

Identifies whether the system enforces and tracks quota limits. All quotas have automatic usage tracking. You can optionally enable enforcement for each quota.

- If RestrictUsers=AllAggregate, enable this option so that the Pillar Axiom system enforces all quota limits in the specified directory.
- If RestrictUsers=SingleUser or Group, disable this option. The system tracks and enforces the user’s or group’s quota limits in the filesystem.

Username [optional]

Identifies a user that the quota covers. Enter a user name, user ID (UID), CIFS account name, or NFS account name.

- If RestrictUsers=SingleUser, you must enter a username.

Groupname [optional]

Identifies a group of users that the quota covers. Enter a group name or group ID (GID).

- If RestrictUsers=Group, you must enter a group name.

SoftLimitInMB

Identifies a soft capacity limit, which means that once the system reaches this value, it sends an administrator alert. If space is available, the system stores data until it reaches the grace period or the hard limit.

Be aware that:

- If a parent directory’s quota limit is smaller than a child directory’s, the system enforces the parent’s limit.
- Users sometimes delete directories from a filesystem's directory structure. If Directory, Directory & User, or Directory & Group quotas are assigned to the deleted directory, the quotas are also deleted.

To define an unlimited value, enter 0 (zero).

HardLimitInMB

Identifies a hard capacity limit, which means that once the system reaches this value, it rejects write requests and does not store data.

To define an unlimited value, enter 0 (zero).

MBGracePeriodInDays

Identifies the number of days that the system can exceed a soft limit. When the time expires, the soft limit becomes a hard limit.

To define an unlimited value, enter 0 (zero).

EXAMPLES

Runs the following command to create a quota on the engineering organization’s filesystem.

```
pdscli request CreateQuota @input
```

where input.req contains the following name/value pairs:

```
Request=CreateQuota
AllowFileSystemOffline=false
FileSystem=/Company-XYZ-FS1/EngAtl
Path=/src
RestrictUsers=AllAggregate
EnforceAllLimitsForDirectory=true
SoftLimitInMB=55
HardLimitInMB=75
MBGracePeriodInDays=5
```

See the CreateQuotaResponse.txt file for example response data.

SEE ALSO

ModifyFileSystem on p. 330

ModifyQuota on p. 375

DeleteQuota on p. 186

GetAllQuotas on p. 224

GetQuotaDetails on p. 282

CreateJob on p. 127

ModifyJob on p. 345

CreateSchedule on p. 159

PerformCancelTask on p. 404

Ranges for Option Values on p. 57

NAME **CreateRoute**

DESCRIPTION Creates a secondary network route for the specified File Server. A route is defined as:

> The sequence of hosts, routers, gateways, and other devices that network traffic can take. Pillar Axiom administrators identify at least one gateway for a File Server to use to route messages to other networks.

The settings for the primary route are attributes of the File Server, which you define when you run the CreateFileServer command.

SYNTAX

```
pdscli {request | submit} CreateRoute
   FileServer=GUID-or-FQN-of-File-Server
   DestinationIPAddress=IP-address
   IPNetmask=IP-netmask
   GatewayIPAddress=IP-address
```

OPTIONS

FileServer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the File Server with which the network route is associated.

DestinationIPAddress

Identifies a destination subnet for this network route. All computers in the subnet are reached through the specified gateway.

IPNetmask

Identifies a subnet mask for the range of IP addresses at the route's destination.

GatewayIPAddress

Identifies the IP address that is assigned to the gateway host. The gateway IP address is used to route messages from this network to other networks.

EXAMPLES Run the following command to add a network route to the `Company`-XYZ-FS1 File Server.

```
pdscli request CreateRoute @input
```

where input.req contains the following name/value pairs:

```
Request=CreateRoute
FileServer=/Company-XYZ-FS1
```

```
DestinationIPAddress=172.20.67.0
IPNetmask=255.255.255.0
GatewayIPAddress=172.18.65.110
```

See the CreateRouteResponse.txt file for example response data.

SEE ALSO

CreateFileServer on p. 100

DeleteRoute on p. 188

GetAllRoutes on p. 226

GetRouteDetails on p. 283

Ranges for Option Values on p. 57

NAME **CreateSchedule**

DESCRIPTION Creates a schedule for recurrent or one-time-only tasks. Assign one or more jobs to the schedule to define which tasks the schedule performs.

SYNTAX

```
pdscli {request | submit} CreateSchedule
   Name=string
   StartTime=YYYY-MM-DDTHH:mm:SS.xx+-HH:mm
   LatestStartTime=
      YYYY-MM-DDTHH:mm:SS.xx+-HH:mm
   RepeatInterval=-PxYxMxDTxHxM.xS
```

OPTIONS

Name

Identifies the name of a scheduled operation, which is an action to be performed at the specified time or at regular intervals.

StartTime AND LatestStartTime [optional]

Identifies the date and time at which the Pillar Axiom system starts a scheduled operation. Use the following format for the entry:

YYYY-MM-DDTHH:mm:SS.xx+-HH:mm

where:

- YYYY-MM-DD designates a four-digit year, two-digit month, and two-digit day for the date.
- T is a separator that designates the start of the time portion of the string.
- HH:mm:SS designates hours, minutes, and seconds in values for a 24-hour clock.
- xx designates a fraction of a second, to two decimal places.
- +-HH:mm designates the time zone as an offset from Coordinated Universal Time (UTC) in hours and minutes. Include the + or - prefix as appropriate for your time zone in relation to UTC.

RepeatInterval [optional]

Identifies the recurrence interval for a scheduled operation. Use the following format for the entry:

-PxYxMxDTxHxM.xxxS

where:

- P is a period separator. The - prefix is optional to designate an interval before the date and time literal constants.
- x represents any literal constant, and the value precedes its date and time separators.
- Y, M, and D are Year, Month, and Day separators.
- T is a separator that designates the start of the time portion of the string.

- H, M, and S are Hour, Minute, and fractional Second separators (seconds to three decimal places). Enter literal constants for a 24-hour clock.

Omit separators and their associated literals if they do not apply to the interval that you want to specify.

EXAMPLES

Run the following command to create a schedule that runs a set of jobs on the first of every month for two years.

```
pdscli request CreateSchedule @input
```

where input.req contains the following name/value pairs:

```
Request=CreateSchedule
Name=FirstOfMonth
StartTime=2004-01-01T12:00:00.-08:00
LatestStartTime=2004-01-01T12:15:00.-08:00
RepeatInterval=P2Y
```

See the CreateScheduleResponse.txt file for example response data.

SEE ALSO

CreateJob on p. 127

DeleteSchedule on p. 190

GetAllJobs on p. 220

GetJobDetails on p. 268

Ranges for Option Values on p. 57

NAME **CreateSnapLUN**

DESCRIPTION Creates a Snap LUN, which is defined as:

A point-in-time, read-write copy of a LUN that you can immediately use. Snap LUNs requires no cache and minimal disk storage space.

Snap LUNs point to the original data, and have the same QoS parameters as the source LUN.

Note: Formerly called a Snap LUN, the GUI now refers to this term as Clone LUN, whereas Snap LUN is still used in the command line.

Use this request for frequent, non-disruptive backup and restore operations.

SYNTAX

```
pdscli {request | submit} CreateSnapLUN
   Source=GUID-or-FQN-of-source-LUN
   EnableiSCSIAccess={true | false}
   EnableFibreChannelAcces={true | false}
   Name=string
   LUNNumber=integer-0-through-255
   Active={true | false}
   Mapped={true | false}
   EnableiSCSIAccess={true | false}
   EnableFibreChannelAccess={true | false}
```

OPTIONS

Source

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the source LUN to copy.

Name

Identifies the name that is assigned to a LUN for administrative purposes. LUN names must be unique across the Pillar Axiom system, not just within its associated volume group.

Name is a required parameter only if you want to change the FQN LUN's user-visible name.

LUNNumber

Identifies the unique number that is assigned to a LUN and can be accessed by all SAN hosts if the LUN is not mapped. If the LUN is mapped, the LUN number must be unique to the mapped SAN host.

Note: If you intend to map this LUN to a specified host, do not include the LUNNumber option.

Active [optional]

Identifies whether the Clone LUN is available to the host (active) or is not seen by the host (inactive) when it is created. Inactive Clone LUNs can be changed to active; however, once the Clone LUN is set to active, you cannot change it to inactive. Default is active.

Mapped [optional]

Identifies whether all hosts or only selected hosts may access this LUN.

- Mapped allows only designated hosts to access this LUN.
- Unmapped allows all hosts to access this LUN using the LUN Number.

Note: If you choose Mapped, do not assign a LUNNumber.

EnableiSCSIAccess

Identifies the protocol specification used when creating a SAN LUN. Options include iSCSI, Fibre Channel (FC), or both.

EnableFibreChannelAccess

Identifies the protocol specification used when creating a SAN LUN. Options include iSCSI, Fibre Channel (FC), or both.

EXAMPLES

Run the following command to create an active clone of EngLUN.

```
pdscli request CreateSnapLUN @input
```

where input.req contains the following name/value pairs:

```
Request=CreateSnapLUN
SourceLUN=/EngLUN
EnableiSCSIAccess=false
EnableFibreChannelAccess=true
Name=EngLUN
LUNNumber= 0
Active=true
Mapped=false
```

See the CreateSnapLUNResponse.txt file for example response data.

SEE ALSO

DeleteSnapLUN on p. 191

GetAllSnapLUNs on p. 233

GetAllSnapLUNHierarchies on p. 232

GetSnapLUNDetails on p. 297

ModifySnapLUN on p. 381

PerformBackgroundLUNDuplication on p. 402

PerformLUNDuplication on p. 447

NAME

CreateVirtualInterface

DESCRIPTION

Creates a secondary virtual network interface for the specified File Server. A virtual interface is defined as:

> A logical interface for regulating network I/O across different processes that access the same physical interface. A virtual network interface:
>
> - Has an IP address that is unique for the File Server.
> - Has a VLAN tag ID that is unique for the File Server and is non-zero if tagging is enabled.
> - Identifies a particular network port on a particular control unit (CU) of a Slammer.
>
> The Pillar Axiom system creates CIFS and NFS servers on both CUs of the associated Slammer.

The settings for the primary virtual interface are attributes of the File Server, which you define when you run the CreateFileServer command.

SYNTAX

```
pdscli {request | submit} CreateVirtualInterface
   FileServer=GUID-or-FQN-of-File-Server
   IPAddress=IP-address
   IPNetmask=IP-netmask
   VLANTag={0 | integer-1-through-4095}
   FrameSize=integer
   NetworkInterfaceIOPort.Slammer=
      GUID-or-FQN-of-Slammer
   NetworkInterfaceIOPort.ControlUnitNumber=
      {0 | 1}
   NetworkInterfaceIOPort.EthernetPortNamer=
      {PORT0 | PORT1}
```

OPTIONS

FileServer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name.

Enter the GUID or FQN of the File Server with which the virtual interface is associated.

IPAddress

Identifies the IP address that is assigned to the File Server’s virtual network interface.

IPNetmask

Identifies the subnet mask that is assigned to the File Server's virtual network interface.

VLANTag [optional]

Identifies the virtual LAN (VLAN) ID that is assigned to the File Server's virtual network interface.

- Assign a value of 1 through 4094, inclusive, if you have connected a VLAN-capable switch to the Pillar Axiom system.
- Leave the field blank to disable VLAN tagging.

FrameSize [optional]

Identifies the packet frame size. This value defines the maximum transmission unit (MTU).

The frame size (MTU) does not include the Ethernet header portion of the packet. If your network switch has trouble with this, you can set the switch to a larger value or lower the MTU size to correct the problem.

- If your network supports extended Ethernet (jumbo) frames, enter an integer greater than 1500 and less than 16362. Make sure that this Pillar Axiom MTU size matches the network MTU size. If the MTU sizes are mismatched, users may experience I/O “hangs” when the client machines try to process too-large packets.
- If your network does not support jumbo frames, enter the default frame size of 1500.

NetworkInterfaceIOPort.Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Slammer that provides the File Server's virtual network interface.

NetworkInterfaceIOPort.ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

NetworkInterfaceIOPort.EthernetPortName

Identifies the port on the Slammer that is assigned to the File Server's virtual network interface. Select a gigabit Ethernet network port on on one of the data-path Slammers control unit (CU) .

- If Link Aggregation is enabled, select port 0 (zero). You cannot create virtual interfaces on alternate port 1 when Link Aggregation is enabled.
- If Link Aggregation is disabled, select either port.

EXAMPLES

Run the following command to create a secondary virtual network interface for the `Company`-XYZ-FS1 File Server.

```
pdscli request CreateVirtualInterface @input
```

where input.req contains the following name/value pairs:

```
Request=CreateVirtualInterface
FileServer=/Company-XYZ-FS1
IPAddress=172.18.75.142
IPNetmask=255.255.255.0
VLANTag=42
FrameSize=1500
NetworkInterfaceIOPort.Slammer=Slammer2
NetworkInterfaceIOPort.ControlUnitNumber=1
NetworkInterfaceIOPort.EthernetPortName=PORT0
```

See the CreateVirtualInterfaceResponse.txt file for example response data.

SEE ALSO

CreateFileServer on p. 100

DeleteVirtualInterface on p. 192

GetAllVirtualInterfaces on p. 239

GetVirtualInterfaceDetails on p. 307

Ranges for Option Values on p. 57

NAME

CreateVolumeGroup

DESCRIPTION

Creates a new volume group to organize logical volumes (filesystems and LUNs).

SYNTAX

```
pdscli {request | submit} CreateVolumeGroup
  Name=volume-group-name
  ParentVolumeGroup=GUID-or-FQN-of-parent-
    volume-group
  MaximumCapacity=integer
```

OPTIONS

Name

Identifies the name that is assigned to a volume group. Each volume group name must be unique within its parent volume group.

ParentVolumeGroup

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the next-highest volume group in the hierarchy. If volume groups are not nested, or if the next-highest level is the System itself, specify the System.

MaximumCapacity [optional]

Identifies the maximum capacity for the volume group. The maximum capacity of the filesystems and nested volume groups that are associated with the volume group cannot exceed this value.

A value of 0 (zero) identifies that the volume group is configured with unlimited capacity. You can increase the maximum capacity of associated filesystems and nested volume groups without constraints.

EXAMPLES

Run the following command to create a volume group for your engineering organization's logical volumes.

```
pdscli request CreateVolumeGroup @input
```

where input.req contains the following name/value pairs:

```
Request=CreateVolumeGroup
Name=/Eng
ParentVolumeGroup=/
MaximumCapacity=0
```

See the CreateVolumeGroupResponse.txt file for example response data.

SEE ALSO

ModifyVolumeGroup on p. 389

DeleteVolumeGroup on p. 195

Ranges for Option Values on p. 57

NAME **DeleteAccount**

DESCRIPTION Deletes the specified administrator account.

SYNTAX

```
pdscli {request | submit} DeleteAccount
   Account=GUID-or-FQN-of-account
```

OPTIONS

Account

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the account to delete.

EXAMPLES Run the following command to delete Yvette Smith's account after she quits her job.

```
pdscli request DeleteAccount @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteAccount
Account=IDf1234567-89ab-cde0-0123-
   456789eeeee0
```

See the DeleteAccountResponse.txt file for example response data.

SEE ALSO

GetAllAccounts on p. 200

GetAccountDetails on p. 198

Ranges for Option Values on p. 57

NAME **DeleteAdministratorAction**

DESCRIPTION Deletes the specified administrator action. Run this command after you perform the corrective action that is suggested in the administrator action.

SYNTAX

```
pdscli {request | submit} DeleteAdministratorAction
   AdministratorAction=GUID-or-FQN-of-action
```

OPTIONS **AdministratorAction**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the administrator action to delete.

EXAMPLES You would like to delete the action that notified you about filesystem inconsistency. Delete the administrator action after you run the filesystem check (FSCK) utility. Run the following command.

```
pdscli request DeleteAdministratorAction @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteAdministratorAction
AdministratorAction=IDf1234567-89ab-cde0-
   0123-456789eeeee0
```

See the DeleteAdministratorActionResponse.txt file for example response data.

SEE ALSO GetAllAdministratorActions on p. 201

GetAdministratorActionDetails on p. 199

Ranges for Option Values on p. 57

NAME

DeleteCIFSShare

DESCRIPTION

Deletes the specified Common Internet File System (CIFS) share.

SYNTAX

```
pdscli {request | submit} DeleteCIFSShare
   CIFSShare=GUID-or-FQN-of-CIFS-share
```

OPTIONS

CIFSShare

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the CIFS share to delete.

EXAMPLES

Run the following command to delete an unused share that was created for the EngAtl filesystem.

```
pdscli request DeleteCIFSShare @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteCIFSShare
CIFSShare=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the DeleteCIFSShareResponse.txt file for example response data.

SEE ALSO

GetAllCIFSShares on p. 206

GetCIFSShareDetails on p. 244

Ranges for Option Values on p. 57

NAME

DeleteCloneFS

DESCRIPTION

Deletes the specified Clone FS of a filesystem.

A Clone FS is a point-in-time, writeable snapshot of a filesystem, which can be used later to revert the filesystem. This snapshot retains the same QoS parameters as the source filesystem. It consumes storage capacity from the Clone FS storage space created for the source filesystem. A Clone FS can be scheduled to occur at any time. This provides a convenient method to branch from the source data without the need to do a full block-level copy.

SYNTAX

```
pdscli {request | submit} DeleteCloneFS
   CloneFSFQN=FQN-of-Clone-FS
   ForceFileRetentionDelete={true | false}
```

OPTIONS

CloneFSFQN

Identifies a specific Pillar Axiom object.

- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

  FQN Format:
  `/Fully/Qualified/ObjectName`

  Enter the FQN of the filesystem with which this share is associated. If there is more than one file server, use the FQN that includes the file server and filesystem.

Enter the GUID or FQN of the Clone FS to delete.

ForceFileRetentionDelete

Forces the deletion of all file retention data for the specified Clone FS of a filesystem.

EXAMPLES

Run the following command to delete a Clone FS that was created from the EngAtl filesystem.

```
pdscli request DeleteCloneFS @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteCloneFS
   CloneFSFQN=/Company-XYZ-FS1/EngAtl
   ForceFileRetentionDelete=true
```

See the DeleteCloneFS.txt file for example response data.

SEE ALSO

CreateCloneFS on p. 97

GetCloneFSDetails on p. 245

GetAllSnapLUNs on p. 233

GetAllCloneFSHierarchies on p. 207

GetAllCloneFSs on p. 208

NAME **DeleteExternalUPSDevice**

DESCRIPTION Deletes the specified uninterruptable power supply.

SYNTAX

```
pdscli {request | submit} DeleteExternalUPSDevice
   ups=GUID-or-FQN-of-UPS-Device
```

OPTIONS

ups

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the UPS device to modify.

EXAMPLES Run the following command to delete a UPS device.

```
pdscli request DeleteExternalUPS Device @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteExternalUPSDevice
ups=UPS1

See the DeleteExternalUPSDeviceResponse.txt file for example response
data.
```

SEE ALSO GetAllExternalUPSDevices on p. 212

Ranges for Option Values on p. 57

NAME **DeleteFileServer**

DESCRIPTION Deletes the specified File Server.

SYNTAX

```
pdscli {request | submit} DeleteFileServer
  FileServer=GUID-or-FQN-of-File-Server
```

OPTIONS **FileServer**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the File Server to delete.

EXAMPLES Run the following command to delete a File Server that is no longer required.

```
pdscli request DeleteFileServer @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteFileServer
FileServer=/Company-XYZ-FS3
```

See the DeleteFileServerResponse.txt file for example response data.

SEE ALSO GetAllFileServers on p. 213

GetFileServerDetails on p. 253

GetFileServerStatistics on p. 254

Ranges for Option Values on p. 57

NAME **DeleteFileSystem**

DESCRIPTION Deletes the specified filesystem.

SYNTAX

```
pdscli {request | submit} DeleteFileSystem
   ID=GUID-or-FQN-of-filesystem
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the filesystem to delete.

EXAMPLES Run the following command to delete the OpsLA filesystem.

```
pdscli request DeleteFileSystem @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteFileSystem
ID=/Company-XYZ-FS1/OpsLA
```

See the DeleteFileSystemResponse.txt file for example response data.

SEE ALSO GetAllFileSystems on p. 214

GetFileSystemDetails on p. 255

GetFileSystemStatistics on p. 260

Ranges for Option Values on p. 57

SNAME **DeleteFileSystemSnapshot**

DESCRIPTION Deletes the specified Snap FS of a filesystem.

SYNTAX

```
pdscli {request | submit} DeleteFileSystemSnapshot
   ID=GUID-or-FQN-of-Snap-FS
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Snap FS to delete.

EXAMPLES Run the following command to delete a Snap FS that was created from the EngAtl filesystem.

```
pdscli request DeleteFileSystemSnapshot @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteFileSystemSnapshot
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the DeleteFileSystemSnapshotResponse.txt file for example response data.

SEE ALSO CreateFileSystemSnapshotSchedule on p. 124

DeleteFileSystemSnapshotSchedule on p. 179

GetAllFileSystemSnapshots on p. 216

GetAllFileSystemSnapshotSchedules on p. 218

GetFileSystemSnapshotDetails on p. 256

GetFileSystemSnapshotScheduleDetails on p. 258

ModifyFileSystemSnapshot on p. 340

PerformFileSystemDuplication on p. 425

PerformBackgroundFileSystemDuplication on p. 400

PerformRestoreFileSystemFromSnapshot on p. 485

PerformRebuildFileSystemRedundancy on p. 474

Ranges for Option Values on p. 57

NAME **DeleteFileSystemSnapshotSchedule**

DESCRIPTION Deletes the specified schedule that creates Snap FSs of a filesystem at designated intervals.

SYNTAX

```
pdscli {request | submit}
  DeleteFileSystemSnapshotSchedule
    ID=GUID-or-FQN-of-Snap-FS-schedule
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the schedule to delete.

EXAMPLES Run the following command to delete a schedule that creates Snap FSs of the EngAtl filesystem.

```
pdscli request DeleteFileSystemSnapshotSchedule
  @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteFileSystemSnapshotSchedule
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the DeleteFileSystemSnapshotSchedule Response.txt file for example response data.

SEE ALSO CreateFileSystemSnapshotSchedule on p. 124

DeleteFileSystemSnapshot on p. 177

GetAllFileSystemSnapshots on p. 216

GetAllFileSystemSnapshotSchedules on p. 218

GetFileSystemSnapshotDetails on p. 256

GetFileSystemSnapshotScheduleDetails on p. 258

ModifyFileSystemSnapshot on p. 340

PerformFileSystemDuplication on p. 425

PerformBackgroundFileSystemDuplication on p. 400

PerformRestoreFileSystemFromSnapshot on p. 485

PerformRebuildFileSystemRedundancy on p. 474

Ranges for Option Values on p. 57

NAME **DeleteJob**

DESCRIPTION Deletes the specified job. A *job* is a management task that can be scheduled in a Pillar Axiom storage system. Not all management tasks can be scheduled; jobs are a subset of all possible management tasks.

SYNTAX

```
pdscli {request | submit} DeleteJob
  Job=GUID-or-FQN-of-scheduled-operation
```

OPTIONS **Job**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the job to delete.

EXAMPLES Run the following command to delete a job that would have created a Snap FS of the EngAtl filesystem.

```
pdscli request DeleteJob @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteJob
Job=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the DeleteJobResponse.txt file for example response data.

SEE ALSO GetAllJobs on p. 220

GetJobDetails on p. 268

Ranges for Option Values on p. 57

NAME **DeleteLUN**

DESCRIPTION Deletes the specified LUN.

SYNTAX

```
pdscli {request | submit} DeleteLUN
   ID=GUID-or-FQN-of-LUN
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the LUN to delete.

EXAMPLES Run the following command to delete a LUN that is not in active use.

```
pdscli request DeleteLUN @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteLUN
ID=/XYZ-FS1/EngLUN
```

See the DeleteLUNResponse.txt file for example response data.

SEE ALSO CreateLUN on p. 135

ModifyLUN on p. 351

GetAllAvailableLUNNumbers on p. 202

GetAllLUNs on p. 221

GetLUNDetails on p. 269

GetLUNStatistics on p. 272

GetHostMappingConfigDetails on p. 265

GetLUNMappingConfigDetails on p. 270

PerformConfigureLUNMapping on p. 412

PerformRebuildLUNRedundancy on p. 475

Ranges for Option Values on p. 57

NAME

DeleteNFSExport

DESCRIPTION

Deletes the specified Network File System (NFS) export and any host entries that are associated with the export.

SYNTAX

```
pdscli {request | submit} DeleteNFSExport
  NFSExport=GUID-or-FQN-of-NFS-export
```

OPTIONS

NFSExport

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the NFS export to delete.

EXAMPLES

Run the following command to delete an unused export that was created for the EngAtl filesystem.

```
pdscli request DeleteNFSExport @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteNFSExport
NFSExport=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the DeleteNFSExportResponse.txt file for example response data.

SEE ALSO

GetAllNFSExports on p. 222

GetNFSExportDetails on p. 276

Ranges for Option Values on p. 57

NAME **DeleteNFSHostEntry**

DESCRIPTION Deletes the specified Network File System (NFS) host entry, which removes the ability of that NFS client (user) to mount the export.

SYNTAX

```
pdscli {request | submit} DeleteNFSHostEntry
   NFSHostEntry=GUID-or-FQN-of-NFS-host-entry
```

OPTIONS **NFSHostEntry**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the NFS host entry to delete.

EXAMPLES Run the following command to delete a host entry from an NFS export.

```
pdscli request DeleteNFSHostEntry @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteNFSHostEntry
NFSHostEntry=IDA0479DC2-38EF-9828-8902-
   E0FC03743270
```

See the DeleteNFSHostEntryResponse.txt file for example response data.

SEE ALSO GetAllNFSExports on p. 222

Ranges for Option Values on p. 57

NAME **DeleteNotificationDefinition**

DESCRIPTION Deletes the specified notification.

SYNTAX

```
pdscli {request | submit}
  DeleteNotificationDefinition
    NotificationDefinition=GUID-or-FQN-of-
      notification
```

OPTIONS **NotificationDefinition**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the notification to delete.

EXAMPLES Run the following command to delete a notification.

```
pdscli request DeleteNotificationDefinition @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteNotificationDefinition
NotificationDefinition=IDA0479DC2-38EF-9828-
  8902-E0FC03743270
```

See the DeleteNotificationDefinitionResponse.txt file for example response data.

SEE ALSO GetAllNotificationDefinitions on p. 223

GetNotificationDefinitionDetails on p. 278

Ranges for Option Values on p. 57

NAME **DeleteQuota**

DESCRIPTION Deletes the specified filesystem quota.

SYNTAX

```
pdscli {request | submit} DeleteQuota
   ID=GUID-or-FQN-of-quota
   AllowFileSystemOffline={true | false}
```

OPTIONS

ID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the quota to delete.

AllowFileSystemOffline

Identifies whether the Pillar Axiom system can take the filesystem offline to perform quota operations.

- Enable this option if the Pillar Axiom system can take the filesystem offline to create or remove a directory-level quota when the specified directory contains files and subdirectories.
- Disable this option if the system cannot take the filesystem offline to perform the quota operation when the specified directory contains files and subdirectories. The filesystem remains online, and the directory-level quota is not created or removed.

Whichever setting you choose, the filesystem remains online and the quota operation is performed when the specified directory is empty.

EXAMPLES Run the following command to delete a quota from the OpsLA filesystem. You know that usage of that filesystem is light, so set the flag to permit the system to take the non-empty filesystem offline.

```
pdscli request DeleteQuota @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteQuota
ID=/Company-XYZ-FS1/OpsLA:/mfg
AllowFileSystemOffline=true
```

See the DeleteQuotaResponse.txt file for example response data.

SEE ALSO GetAllQuotas on p. 224

GetQuotaDetails on p. 282

GetAllTasks on p. 235

PerformCancelTask on p. 404

Ranges for Option Values on p. 57

NAME **DeleteRoute**

DESCRIPTION Deletes the specified secondary network route. File Servers have a primary network route and optional secondary routes. The primary network route is an attribute of the File Server; it is not a separate object that you can delete.

SYNTAX

```
pdscli {request | submit} DeleteRoute
   Route=GUID-or-FQN-of-route
```

OPTIONS **Route**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the route to delete.

EXAMPLES Run the following command to delete a secondary network route from the Company-XYZ-FS1 File Server.

```
pdscli request DeleteRoute @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteRoute
Route=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the DeleteRouteResponse.txt file for example response data.

SEE ALSO GetAllRoutes on p. 226

GetRouteDetails on p. 283

Ranges for Option Values on p. 57

NAME **DeleteSANHost**

DESCRIPTION Deletes a specified SAN host from your SAN configuration.

SYNTAX

```
pdscli {request | submit} DeleteSANHost
   SANHostName=Host-name
```

OPTIONS **SANHostName**

Identifies the name assigned to a specific SAN host.

EXAMPLES Run the following command to delete the SANTest_03 host.

```
pdscli request DeleteSANHost@input
```

where input.req contains the following name/value pairs:

```
Request=DeleteSANHost
SANHostName=SANTest_03
```

See the DeleteSANHostResponse.txt file for example response data.

SEE ALSO GetSANConfigDetails on p. 284

GetHostMappingConfigDetails on p. 265

GetAllSANHosts on p. 227

ModifySANHost on p. 377

GetHBAPortConnections on p. 262

PerformAssociateInitiatorsToHost on p. 398

Ranges for Option Values on p. 57

NAME

DeleteSchedule

DESCRIPTION

Deletes the specified schedule. A schedule defines when the Pillar Axiom storage system performs the jobs that are associated with the schedule.

SYNTAX

```
pdscli {request | submit} DeleteSchedule
   Schedule=GUID-or-FQN-of-schedule
```

OPTIONS

Schedule

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the schedule to delete.

EXAMPLES

Run the following command to delete a schedule that has no jobs associated with it.

```
pdscli request DeleteSchedule @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteSchedule
Schedule=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the DeleteScheduleResponse.txt file for example response data.

SEE ALSO

GetAllSchedules on p. 229

GetScheduleDetails on p. 291

Ranges for Option Values on p. 57

NAME **DeleteSnapLUN**

DESCRIPTION Deletes a specified Snap LUN.

.SYNTAX

```
pdscli {request | submit} DeleteSnapLUN
  ID=GUID-or-FQN-of-Snap-LUN
```

OPTIONS

ID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Snap LUN to delete.

EXAMPLES Run the following command to delete SLUN_01.

```
pdscli request DeleteSnapLUN@input
```

where input.req contains the following name/value pairs:

```
Request=DeleteSnapLUN
ID=/SLUN_01
```

See the `DeleteSnapLUN`Response.txt file for example response data.

SEE ALSO

GetAllTapeRobots on p. 234

GetAllSnapLUNHierarchies on p. 232

GetSnapLUNDetails on p. 297

ModifySnapLUN on p. 381

PerformLUNDuplication on p. 447

PerformBackgroundLUNDuplication on p. 402

Ranges for Option Values on p. 57

NAME

DeleteVirtualInterface

DESCRIPTION

Deletes the specified secondary virtual interface. File Servers have a primary virtual interface and optional secondary virtual interfaces. The primary virtual interface is an attribute of the File Server; it is not a separate object that you can delete.

SYNTAX

```
pdscli {request | submit} DeleteVirtualInterface
  VirtualInterface=GUID-or-FQN-of-interface
```

OPTIONS

VirtualInterface

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the virtual network interface to delete.

EXAMPLES

Run the following command to delete a secondary virtual interface from the `Company`-XYZ-FS1 File Server.

```
pdscli request DeleteVirtualInterface @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteVirtualInterface
VirtualInterface=IDA0479DC2-38EF-9828-8902-
    E0FC03743270
```

See the DeleteVirtualInterfaceResponse.txt file for example response data.

SEE ALSO

GetAllVirtualInterfaces on p. 239

GetVirtualInterfaceDetails on p. 307

Ranges for Option Values on p. 57

NAME **DeleteVolumeAndChildren**

DESCRIPTION Deletes the specified volume and all of its clones, provided that the following conditions are met:

- This task can be run on any type of volume or clone, active or inactive, unless there are incomplete background copies of a clone in progress somewhere in the clone tree, or an upstream volume is being restored from a clone.
- If a restore is in progress, the request to delete the volume and its children is allowed only if the volume is the topmost volume. That is, unless the volume being deleted is also the one being restored, the request is rejected.
- Data migrations do not count as background copies in this case. If a data migration is in progress on a volume being deleted, the deletion proceeds to clean up the migration data.

SYNTAX

```
pdscli {request | submit} DeleteVolumeAndChildren
   CloneFSID=GUID-of-CloneFS | CloneFSFQN=/FQN-of-CloneFS |
   FileSystemID=GUID-of-FileSystem | FileSystemFQN=/FQN-of-FileSystem
   |
   LUNID=GUID-of-LUN | LUNFQN=/FQN-of-LUN |
   CloneLUNID=GUID-of-CloneLUN | CloneLUNFQN=/FQN-of-CloneLUN
   [ ForceFileRetentionDelete={ true | false } ]
```

OPTIONS ***Volume*ID or *Volume*FQN**

Volume identifies an object of a specific type. Choose one of:

- CloneFS
- CloneLUN
- FileSystem
- LUN

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the volume tree to delete.

ForceFileRetentionDelete

Identifies whether deletion of a non-compliant SecureWORMfs is allowed, even if it contains immutable files.

EXAMPLES

Run the following command to delete a volume tree.

```
pdscli request DeleteVolumeAndChildren @input
```

where input.req contains the following name-value pairs:

```
Request=DeleteVolumeAndChildren
CloneFSID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the DeleteVolumeAndChildrenResponse.txt file for example response data.

SEE ALSO

DeleteFileSystem on p. 176

DeleteLUN on p. 182

NAME

DeleteVolumeGroup

DESCRIPTION

Deletes the specified volume group.

SYNTAX

```
pdscli {request | submit} DeleteVolumeGroup
   VolumeGroup=GUID-or-FQN-of-volume-group
```

OPTIONS

VolumeGroup

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the volume group to delete.

EXAMPLES

Run the following command to delete a volume group.

```
pdscli request DeleteVolumeGroup @input
```

where input.req contains the following name/value pairs:

```
Request=DeleteVolumeGroup
VolumeGroup=IDA0479DC2-38EF-9828-8902-
      E0FC03743270
```

See the DeleteVolumeGroupResponse.txt file for example response data.

SEE ALSO

GetAllVolumeGroups on p. 240

GetVolumeGroupDetails on p. 308

Ranges for Option Values on p. 57

NAME **ForceFileSystemRebalanceOnRestart**

DESCRIPTION Sets a flag that instructs the restart process to reassign all auto-assigned filesystems across all the NAS Slammer control units (CUs) present in the system. The restart process does not remove this flag. The flag can be removed only by the CancelForceFileSystemRebalance request

If another restart should happen while this flag is still set, filesystem CU ownership is reassigned again.

SYNTAX

```
pdscli {request | submit} ForceFileSystemRebalanceOnRestart
```

OPTIONS None

EXAMPLES Run the following command to force the reassignment of filesystems across all NAS Slammer CUs.

```
pdscli request ForceFileSystemRebalanceOnRestart
```

See the ForceFileSystemRebalanceOnRestart.txt file for example response data.

SEE ALSO CancelForceFileSystemRebalance on p. 91

NAME **GatherQuotaForFileSystem**

DESCRIPTION Generates a quota report for filesystems and Clone FSs. You must run this request before running the GetAllQuotas request. Doing so provides information that can be returned by the latter.

SYNTAX

```
pdscli {request | submit} GatherQuotaForFileSystem
  FileSystem=GUID-or-FQN
```

OPTIONS

FileSystem

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem or Clone FS for which you want to generate a quota report.

EXAMPLES Run the following command to collect quota information for a specific filesystem and to place that information into a file. The contents of this file can be retrieved at a later time by issuing a GetAllQuotas request.

```
pdscli request GatherQuotaForFileSystem @input
```

where input.req contains the following name/value pairs:

```
Request=GatherQuotaForFileSystem
FileSystem=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GatherQuotaForFileSystemResponse.txt file for example response data.

SEE ALSO

CreateQuota on p. 154

DeleteQuota on p. 186

GetAllQuotas on p. 224

GetAllQuotaFiles on p. 225

GetQuotaDetails on p. 282

ModifyQuota on p. 375

NAME **GetAccountDetails**

DESCRIPTION Displays information about the specified administrator accounts.

SYNTAX

```
pdscli {request | submit} GetAccountDetails
   Account=GUID-or-FQN-of-account
```

OPTIONS **Account**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all accounts for which you want to display information.

EXAMPLES Run the following command to display information about Yvette Smith's account.

```
pdscli request GetAccountDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetAccountDetails
Account=IDf1234567-89ab-cde0-0123-456789eeeee0
```

See the GetAccountDetailsResponse.txt file for example response data.

SEE ALSO GetAllAccounts on p. 200

Ranges for Option Values on p. 57

NAME **GetAdministratorActionDetails**

DESCRIPTION Displays information about the specified administrator action that has not yet been resolved.

SYNTAX

```
pdscli {request | submit}
  GetAdministratorActionDetails
    AdministratorAction=GUID-or-FQN-of-
      action
```

OPTIONS **AdministratorAction**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all administrator actions for which you want to display information.

EXAMPLES Run the following command to display information about an unresolved administrator action.

```
pdscli request GetAdministratorActionDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetAdministratorActionDetails
AdministratorAction=IDf1234567-89ab-cde0-
  0123-456789eeeee0
```

See the GetAdministratorActionDetailsResponse.txt file for example response data.

SEE ALSO GetAllAdministratorActions on p. 201

Ranges for Option Values on p. 57

NAME **GetAllAccounts**

DESCRIPTION Displays a list of administrator accounts.

SYNTAX

```
pdscli {request | submit} GetAllAccounts
```

OPTIONS None.

EXAMPLES Run the following command to display the currently defined administrator accounts.

```
pdscli request GetAllAccounts
```

See the GetAllAccountsResponse.txt file for example response data.

SEE ALSO GetAccountDetails on p. 198

Ranges for Option Values on p. 57

NAME **GetAllAdministratorActions**

DESCRIPTION Displays a list of unresolved administrator actions in the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit}
   GetAllAdministratorActions
```

OPTIONS None.

EXAMPLES Run the following command to find out if he needs to fix any unresolved administrator actions.

```
pdscli request GetAllAdministratorActions
```

See the GetAllAdministratorActionsResponse.txt file for example response data.

SEE ALSO GetAdministratorActionDetails on p. 199

DeleteAdministratorAction on p. 170

Ranges for Option Values on p. 57

NAME **GetAllAvailableLUNNumbers**

DESCRIPTION Displays a list of all available LUN numbers for a specified SAN host.

SYNTAX

```
pdscli {request | submit} GetAllAvailableLUNNumbers
   SANHostName=Host-name
```

OPTIONS **SANHostName**

Identifies the name assigned to a specific SAN host.

EXAMPLES Run the following command to display a list of available LUN numbers for the SANTest_02 host.

```
pdscli request GetAllAvailableLUNNumbers @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllAvailableLUNNumbers
SANHostName=SANTest_02
```

See the GetAllAvailableLUNNumbersForHostResponse.txt file for example response data.

SEE ALSO GetAllSANHosts on p. 227

Ranges for Option Values on p. 57

NAME

GetAllBackupRuns

DESCRIPTION

Displays a list of system backups that occurred during the specified time period.

SYNTAX

```
pdscli {request | submit} GetAllBackupRuns
  TimeFilter.AfterTimestamp=
    YYYY-MM-DDTHH:mm:SS.xx+-HH:mm
  TimeFilter.BeforeTimestamp=
    YYYY-MM-DDTHH:mm:SS.xx+-HH:mm
```

OPTIONS

TimeFilter.AfterTimestamp AND TimeFilter.BeforeTimestamp

Identifies the time period for data collection. Use the following format for the entry:

YYYY-MM-DDTHH:mm:SS.xx+-HH:mm

where:

- YYYY-MM-DD designates a four-digit year, two-digit month, and two-digit day for the date.
- T is a separator that designates the start of the time portion of the string.
- HH:mm:SS designates hours, minutes, and seconds in values for a 24-hour clock.
- xx designates a fraction of a second, to two decimal places.
- +-HH:mm designates the time zone as an offset from Coordinated Universal Time (UTC) in hours and minutes. Include the + or - prefix as appropriate for your time zone in relation to UTC.

EXAMPLES

Run the following command to display a list of backup runs that were performed in June 2004.

```
pdscli request GetAllBackupRuns @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllBackupRuns
TimeFilter.AfterTimestamp=
  2004-06-01T00:01:00.00-08:00
TimeFilter.BeforeTimestamp=
  2004-06-30T00:59:59.59-08:00
```

See the GetAllBackupRunsResponse.txt file for example response data.

SEE ALSO

GetBackupRunStatistics on p. 241

Ranges for Option Values on p. 57

NAME **GetAllBricks**

DESCRIPTION Displays a list of Bricks that are configured on the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetAllBricks
```

OPTIONS None.

EXAMPLES Run the following command to display a list of the Bricks in the Pillar Axiom storage system.

```
pdscli request GetAllBricks
```

See the GetAllBricksResponse.txt file for example response data.

SEE ALSO GetBrickDetails (p. 242)

Ranges for Option Values on p. 57

NAME **GetAllCallHomeFileDetails**

DESCRIPTION Displays results from the most recent Call-Home messages that have been sent to the Pillar World Wide Customer Support Center.

The Call-Home feature is defined as:

A Pillar Axiom feature that, when enabled, allows the system to notify the Pillar World Wide Customer Support Center of critical issues specific to the Pillar Axiom storage system system. No customer data is transmitted. The Call-Home feature transfers files over the Internet using one of the following user-selected methods:

- SCP: Uses secure copy (SCP) with 1024-bit encryption and secure keys.
- HTTPS: Sends files either directly to Pillar World Wide Customer Support Center or through a proxy server for security purposes. Can also be used when the Pillar Axiom storage system does not have direct access to the Internet.

SYNTAX

```
pdscli {request | submit}
  GetAllCallHomeFileDetails
```

OPTIONS None.

EXAMPLES Run the following command to display results of the latest system configuration verification.

```
pdscli request GetAllCallHomeFileDetails
```

See the GetAllCallHomeFileDetailsResponse.txt file for example response data.

SEE ALSO GetSystemInformationFileDetails on p. 302

PerformCollectSystemInformation on p. 407

PerformDownloadCallHomeFile on p. 418

PerformDownloadSystemInformationFile on p. 420

PerformPeriodicCallHome on p. 460

PerformTestCallHome on p. 505

NAME **GetAllCIFSShares**

DESCRIPTION Displays a list of Common Internet File System (CIFS) shares that are defined for the specified filesystem.

SYNTAX

```
pdscli {request | submit} GetAllCIFSShares
   FileSystem=GUID-or-FQN-of-filesystem
```

OPTIONS **FileSystem**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the filesystem for which you want to display a list of CIFS shares.

EXAMPLES Run the following command to display a list of CIFS shares for the engineering organization's filesystem.

```
pdscli request GetAllCIFSShares @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllCIFSShares
FileSystem=/Company-XYZ-FS1/EngAtl
```

See the GetAllCIFSSharesResponse.txt file for example response data.

SEE ALSO GetCIFSShareDetails on p. 244

Ranges for Option Values on p. 57

NAME **GetAllCloneFSHierarchies**

DESCRIPTION Displays a list of all snapshot hierarchies on a Pillar Axiom system. A storage hierarchy is a range of memory and storage devices within the Pillar Axiom system which uses all storage media to store and retrieve data. The Clone FS Hierarchy is that of all snapshots on the system.

A Clone FS is a point-in-time, writeable snapshot of a filesystem, which can be used later to revert the filesystem. This snapshot retains the same QoS parameters as the source filesystem. It consumes storage capacity from the Clone FS storage space created for the source filesystem. A Clone FS can be scheduled to occur at any time. This provides a convenient method to branch from the source data without the need to do a full block-level copy.

SYNTAX

```
pdscli {request | submit} GetAllCloneFSHierarchies
```

OPTIONS None

EXAMPLES Run the following command to display a list of all Clone FS hierarchies.

```
pdscli request GetAllCloneFSHierarchies @input
```

See the GetAllCloneFSHierarchiesResponse.txt file for example response data

SEE ALSO CreateCloneFS on p. 97

GetCloneFSDetails on p. 245

GetAllSnapLUNs on p. 233

GetAllCloneFSs on p. 208

GetAllCloneFSHierarchies on p. 207

GetAllSnapLUNHierarchies on p. 232

NAME

GetAllCloneFSs

DESCRIPTION

Displays a list of Clone FSs that have been created from the specified Pillar Axiom object.

A Clone FS is a point-in-time, writeable snapshot of a filesystem, which can be used later to revert the filesystem. This snapshot retains the same QoS parameters as the source filesystem. It consumes storage capacity from the Clone FS storage space created for the source filesystem. A Clone FS can be scheduled to occur at any time. This provides a convenient method to branch from the source data without the need to do a full block-level copy.

SYNTAX

```
pdscli {request | submit} GetAllCloneFSs
   FileServerFQN=FQN-of-File-Server
   FileSystemFQN=FQN-of-filesystem
   VolumeGroupFQN=FQN-of-volume-group
   FileRetentionFilter=FileRetentionOnly
```

OPTIONS

FileServerFQN, FileSystemFQN, VolumeGroupFQN

Identifies a specific Pillar Axiom object.

- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name.

  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the FQN of the filesystem with which this share is associated. If there is more than one file server, use the FQN that includes the file server and filesystem.

FileRetentionFilter

Specifies a list of Clone FSs that are used for file retention only.

EXAMPLES

Run the following command to list the Clone FSs that have been created from the EngAtl filesystem.

```
pdscli request GetAllCloneFSs @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllCloneFSs
   FileServerFQN=/Company-XYZ-FS1/EngAtl
   FileSystemFQN=/Company-XYZ-FSys1/EngAt1
   VolumeGroupFQN=/Company-XYZ-VG1/EngAtl
```

See the GetAllCloneFSs.txt file for example response data.

SEE ALSO

CreateCloneFS on p. 97

GetCloneFSDetails on p. 245

GetAllSnapLUNs on p. 233

GetAllCloneFSs on p. 208

GetAllCloneFSHierarchies on p. 207

NAME **GetAllEvents**

DESCRIPTION Displays a list of system events that match the specified selection criteria.

SYNTAX

```
pdscli {request | submit} GetAllEvents
  TimeFilter.AfterTimestamp=
    YYYY-MM-DDTHH:mm:SS.xx+-HH:mm
  TimeFilter.BeforeTimestamp=
    YYYY-MM-DDTHH:mm:SS.xx+-HH:mm
  SeverityFilter.Severity={Critical | Error
    | Warning | Informational}
  IncludeEventTypeFilter.EventType=string
  ExcludeEventTypeFilter.EventType=string
  FirstReturnedEventNumber=integer
  NumberOfReturnedEvents=integer
```

OPTIONS

TimeFilter.AfterTimestamp AND TimeFilter.BeforeTimestamp [optional]

Identifies the time period for data collection. Use the following format for the entry:

YYYY-MM-DDTHH:mm:SS.xx+-HH:mm

where:

- YYYY-MM-DD designates a four-digit year, two-digit month, and two-digit day for the date.
- T is a separator that designates the start of the time portion of the string.
- HH:mm:SS designates hours, minutes, and seconds in values for a 24-hour clock.
- xx designates a fraction of a second, to two decimal places.
- +-HH:mm designates the time zone as an offset from Coordinated Universal Time (UTC) in hours and minutes. Include the + or - prefix as appropriate for your time zone in relation to UTC.

SeverityFilter.Severity [optional]

Identifies a category of event severities. Choose from:

- Critical, which means that access to data is compromised (for example, two disks in a RAID array have failed) or data loss has occurred.
- Error, which means that administrator action is required to prevent a “hard” error (for example, a single disk in a RAID array has failed) from becoming a critical event.
- Warning, which means that administrator action is required to prevent a “soft” error from becoming an error or critical event.
- Informational, which means that a configuration change has been detected or another non-error event has occurred.

IncludeEventTypeFilter.EventType AND ExcludeEventTypeFilter.EventType[optional]

Identifies the types of events to include or exclude in the response. See the GetAllEventsRequest.req for a list of valid values. This request is available in the CLI's supplemental cliSampleInput subdirectory (see "Extract the pdscli Executable and Supplemental Files" on page 16 for more information).

FirstReturnedEventNumber

Identifies which event in the sequence should be the first one that the Pillar Axiom storage system displays. Event number 1 is the most recent event in the event log.

NumberOfReturnedEvents

Identifies how many events that meet the specified filter criteria should be displayed.

- The FirstReturnedEventNumber value of 1 is the most recent event.
- The NumberOfReturnedEvents will return the events starting with the most recent event. Setting this value to 0 returns all stored events.

EXAMPLES

Run the following command to display a list of critical system events.

```
pdscli request GetAllEvents @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllEvents
SeverityFilter.Severity=Critical
FirstReturnedEventNumber=1
NumberOfReturnedEvents=2500
```

See the GetAllEventsResponse.txt file for example response data.

SEE ALSO

GetAllNotificationDefinitions on p. 223

GetNotificationDefinitionDetails on p. 278

GetSNMPConfigDetails on p. 298

Ranges for Option Values on p. 57

NAME **GetAllExternalUPSDevices**

DESCRIPTION Displays a list of all uninterruptable power supplies.

SYNTAX

```
pdscli {request | submit} GetAllExternalUPSDevices
```

OPTIONS None.

EXAMPLES Run the following command to display a list of external UPS devices.

```
pdscli request GetAllExternalUPSDevices @input
```

where input.req contains the following name/value pairs:

```
Request = GetAllExternalUPSDevices
```

See the GetAllExternalUPSDevicesResponse.txt file for example response data.

SEE ALSO GetAllExternalUPSDevices on p. 212

Ranges for Option Values on p. 57

NAME **GetAllFileServers**

DESCRIPTION Displays a list of File Servers that are configured on the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetAllFileServers
```

OPTIONS None.

EXAMPLES Run the following command to display a list of File Servers that are configured on your system.

```
pdscli request GetAllFileServers
```

See the GetAllFileServersResponse.txt file for example response data.

SEE ALSO GetFileServerDetails on p. 253

Ranges for Option Values on p. 57

NAME **GetAllFileSystems**

DESCRIPTION Displays a list of NAS filesystems that are configured on the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetAllFileSystems
   FileServer=GUID-or-FQN-of-File-Server
   VolumeGroup=GUID-or-FQN-of-volume-group
   FileRetentionFilter={AllRetention | VolumeRetentionOnly |
   FileRetentionOnly}
```

OPTIONS

FileServer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the File Server for which you want to display a list of filesystems.

VolumeGroup

Enter the GUID or FQN of the volume group with which the filesystem is associated.

FileRetentionFilter

Displays specific, retained filesystem information.

- The FileRetentionOnly filter displays a list of all filesystems that have FileRetentionOnly enabled.
- The VolumeRetentionOnly filter displays a list of all filesystems that have VolumeRetentionOnly enabled.
- The AllRetention filter displays the combination of FileRetentionOnly and VolumeRetentionOnly.

EXAMPLES Run the following command to list the filesystems that are associated with the File Server that is named `Company-`XYZ-FS1.

```
pdscli request GetAllFileSystems @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllFileSystems
  FileServer=/Company-XYZ-FS1
  VolumeGroup=/Eng
```

See the GetAllFileSystemsResponse.txt file for example response data.

SEE ALSO GetFileSystemDetails on p. 255

Ranges for Option Values on p. 57

NAME

GetAllFileSystemSnapshots

DESCRIPTION

Displays a list of Snap FSs that have been created from the specified filesystem.

SYNTAX

```
pdscli {request | submit} GetAllFileSystemSnapshots
   FS=GUID-or-FQN-of-filesystem
```

OPTIONS

FS

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the filesystem for which you want to display a list of Snap FSs.

EXAMPLES

Run the following command to list the Snap FSs that have been created from the EngAtl filesystem.

```
pdscli request GetAllFileSystemSnapshots @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllFileSystemSnapshots
FS=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GetAllFileSystemSnapshotsResponse.txt file for example response data.

SEE ALSO

CreateFileSystemSnapshotSchedule on p. 124

DeleteFileSystemSnapshot on p. 177

DeleteFileSystemSnapshotSchedule on p. 179

GetAllFileSystemSnapshotSchedules on p. 218

GetFileSystemSnapshotDetails on p. 256

GetFileSystemSnapshotScheduleDetails on p. 258

ModifyFileSystemSnapshot on p. 340

PerformFileSystemDuplication on p. 425

PerformBackgroundFileSystemDuplication on p. 400

PerformRestoreFileSystemFromSnapshot on p. 485

PerformRebuildFileSystemRedundancy on p. 474

Ranges for Option Values on p. 57

NAME **GetAllFileSystemSnapshotSchedules**

DESCRIPTION Displays a list of schedules that create Snap FSs of the specified filesystem.

SYNTAX

```
pdscli {request | submit}
  GetAllFileSystemSnapshotSchedules
    FS=GUID-or-FQN-of-filesystem
```

OPTIONS **FS**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the filesystem for which you want to display a list of Snap FS schedules.

EXAMPLES Run the following command to list the Snap FS schedules that are defined for the EngAtl filesystem.

```
pdscli request GetAllFileSystemSnapshotSchedules
  @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllFileSystemSnapshotSchedules
FS=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GetAllFileSystemSnapshotSchedules Response.txt file for example response data.

SEE ALSO CreateFileSystemSnapshotSchedule on p. 124

DeleteFileSystemSnapshot on p. 177

DeleteFileSystemSnapshotSchedule on p. 179

GetAllFileSystemSnapshots on p. 216

GetFileSystemSnapshotDetails on p. 256

GetFileSystemSnapshotScheduleDetails on p. 258

ModifyFileSystemSnapshot on p. 340

PerformFileSystemDuplication on p. 425

PerformBackgroundFileSystemDuplication on p. 400

PerformRestoreFileSystemFromSnapshot on p. 485

PerformRebuildFileSystemRedundancy on p. 474

Ranges for Option Values on p. 57

NAME **GetAllJobs**

DESCRIPTION Displays a list of jobs that are associated with the specified schedule and will be performed in the future.

SYNTAX

```
pdscli {request | submit} GetAllJobs
   Schedule=GUID-or-FQN-of-schedule
```

OPTIONS **Schedule**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the schedule for which you want to display a list of jobs.

EXAMPLES Run the following command to display a list of the future jobs for a schedule that creates a Snap FS of the EngAtl filesystem every day.

```
pdscli request GetAllJobs @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllJobs
Schedule=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GetAllJobsResponse.txt file for example response data.

SEE ALSO GetAllTasks on p. 235

GetJobDetails on p. 268

Ranges for Option Values on p. 57

NAME **GetAllLUNs**

DESCRIPTION Displays a list of all LUNs associated with a specified volume group.

SYNTAX

```
pdscli {request | submit} GetAllLUNs
   VolumeGroup=GUID-or-FQN-of-volume-group
```

OPTIONS **VolumeGroup [optional]**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the volume group for which you want to display LUNs. If you do not include the volume group, the system lists all LUNs in all volume groups.

EXAMPLES Run the following command to list the LUNs that are associated with the volume group named Eng.

```
pdscli request GetAllLUNs @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllLUNs
VolumeGroup=/Eng
```

See the GetAllLUNsResponse.txt file for example response data.

SEE ALSO CreateLUN on p. 135

PerformLUNDuplication on p. 447

PerformBackgroundLUNDuplication on p. 402

Ranges for Option Values on p. 57

NAME **GetAllNFSExports**

DESCRIPTION Displays a list of Network File System (NFS) exports that are defined for the specified filesystem.

SYNTAX

```
pdscli {request | submit} GetAllNFSExports
   FileSystem=GUID-or-FQN-of-filesystem
```

OPTIONS

FileSystem

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the filesystem for which you want to display a list of NFS exports.

EXAMPLES Run the following command to display a list of NFS exports for the EngAtl filesystem.

```
pdscli request GetAllNFSExports @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllNFSExports
FileSystem=/Company-XYZ-FS1/EngAtl
```

See the GetAllNFSExportsResponse.txt file for example response data.

SEE ALSO GetNFSExportDetails on p. 276

Ranges for Option Values on p. 57

NAME **GetAllNotificationDefinitions**

DESCRIPTION Displays a list of notifications that are defined for the specified client.

SYNTAX

```
pdscli {request | submit}
  GetAllNotificationDefinitions
    Client={SNMP | Email}
```

OPTIONS **Client**

Identifies the notification client. Choose from:

- SNMP, if the client is a Simple Network Management Protocol (SNMP) application.
- Email, if the client is an electronic mail (email) server that sends the notifications to designated email addresses.

Enter the GUID or FQN of the client for which you want to display a list of notifications.

EXAMPLES Run the following command to display a list of notifications for an email client.

```
pdscli request GetAllNotificationDefinitions @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllNotificationDefinitions
Client=Email
```

See the GetAllNotificationDefinitionsResponse.txt file for example response data.

SEE ALSO GetNotificationDefinitionDetails on p. 278

Ranges for Option Values on p. 57

NAME **GetAllQuotas**

DESCRIPTION Displays a list of quotas, or capacity limits, that are defined for a specified filesystem.

Important! Attempting to retrieve definitions for more than 500 quotas for a specified filesystem is not recommended because of the potential negative impact to the Pilot's performance.

Tip: To obtain the latest quota log, run the GatherQuotaForFilesystem request before running GetAllQuotas.

SYNTAX

```
pdscli {request | submit} GetAllQuotas
   ID=GUID-or-FQN-of-FileSystem
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem for which you want to display a list of quotas.

EXAMPLES Run the following command to display a list of quotas for the EngAtl filesystem.

```
pdscli request GetAllQuotas @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllQuotas
ID=/Company-XYZ-FS1/EngAtl
```

See the GetAllQuotasResponse.txt file for example response data.

SEE ALSO GetQuotaDetails on p. 282

Ranges for Option Values on p. 57

GatherQuotaForFileSystem on p. 197

NAME **GetAllQuotaFiles**

DESCRIPTION Returns a list of available quota report file names, which can be compressed into a quota report and downloaded to the user's client.

SYNTAX

```
pdscli {request | submit} GetAllQuotaFiles
   ID=GUID-or-FQN-of-quota
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem for which you want to generate a quota report.

EXAMPLES Run the following command to display information about a quota that is defined for the engineering organization's filesystem, EngAtl.

```
pdscli request GetAllQuotaFiles @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllQuotaFiles
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GetAllQuotaFilesResponse.txt file for example response data.

NAME **GetAllRoutes**

DESCRIPTION Displays a list of network routes that are defined for the specified File Server.

SYNTAX

```
pdscli {request | submit} GetAllRoutes
   FileServer=GUID-or-FQN-of-File-Server
```

OPTIONS **FileServer**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the File Server for which you want to display a list of routes.

EXAMPLES Run the following command to display a list of routes that are defined for the `Company`-XYZ-FS1 File Server.

```
pdscli request GetAllRoutes @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllRoutes
FileServer=/Company-XYZ-FS1
```

See the GetAllRoutesResponse.txt file for example response data.

SEE ALSO GetAllFileServers on p. 213

GetRouteDetails on p. 283

Ranges for Option Values on p. 57

NAME **GetAllSANHosts**

DESCRIPTION Lists all SAN hosts. The Pillar Axiom storage system detects HBA ports and assigns hosts to those that you do not assign. Use this request to identify all SAN hosts, both user-defined and system-defined.

SYNTAX

```
pdscli {request | submit} GetAllSANHosts
```

OPTIONS None.

EXAMPLES Run the following command to display a list of SAN hosts.

```
pdscli request GetAllSANHosts @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllSANHosts
```

See the GetAllSANHostsResponse.txt file for example response data.

SEE ALSO GetSANConfigDetails on p. 284

GetHostMappingConfigDetails on p. 265

GetAllSANHosts on p. 227

ModifySANHost on p. 377

GetHBAPortConnections on p. 262

PerformAssociateInitiatorsToHost on p. 398

DeleteSANHost on p. 189

Ranges for Option Values on p. 57

NAME **GetAllSANProtocolStatistics**

DESCRIPTION Displays the SAN statistics for all the SAN network interfaces in the Pillar Axiom system.

SYNTAX

```
pdscli {request | submit} GetAllSANProtocolStatistics
```

OPTIONS None.

EXAMPLES Run the following command to display SAN protocol statistics for all SAN interfaces.

```
pdscli request GetAllSANProtocolStatistics @input
```

where input.req contains the following name/value pair:

```
Request = GetAllSANProtocolStatistics
```

See the GetAllSANProtocolStatistics Response.txt file for example response data.

SEE ALSO GetAllSlammers (p. 231)

GetHBAPortConnections (p. 262)

GetiSCSIPortStatisticsDetails (p. 266)

GetSANPortDetails (p. 286)

GetSANProtocolStatistics (p. 288)

GetSANSlammerPortDetails (p. 290)

GetSlammerStatistics (p. 296)

ModifyHBAPort (p. 342)

ModifyiSCSIPortDetails (p. 343)

NAME **GetAllSchedules**

DESCRIPTION Displays a list of schedules that perform jobs at the scheduled time.

SYNTAX

```
pdscli {request | submit} GetAllSchedules
```

OPTIONS None.

EXAMPLES Run the following command to display a list of scheduled operations that are defined on your system.

```
pdscli request GetAllSchedules
```

See the GetAllSchedulesResponse.txt file for example response data.

SEE ALSO GetScheduleDetails on p. 291

GetAllJobs on p. 220

Ranges for Option Values on p. 57

NAME **GetAllSessions**

DESCRIPTION Displays a list of administrator sessions. Every administrator that is logged in has an active session.

SYNTAX

```
pdscli {request | submit} GetAllSessions
```

OPTIONS None.

EXAMPLES Run the following command to list the active administrator sessions.

```
pdscli request GetAllSessions
```

See the GetAllSessionsResponse.txt file for example response data.

SEE ALSO GetAllAccounts on p. 200

GetAccountDetails on p. 198

Ranges for Option Values on p. 57

NAME

GetAllSlammers

DESCRIPTION

Displays a list of Slammers that are configured on the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetAllSlammers
   ServiceType={NAS | SAN}
```

OPTIONS

ServiceType

Identifies whether a Slammer provides NAS or SAN services. Choose from:

- NAS, which specifies that the Slammer is configured with NAS network interface modules and provides NAS services.
- SAN, which specifies that the Slammer is configured with SAN network interface modules and provides SAN services.

EXAMPLES

Run the following command to display a list of NAS Slammers on your system.

```
pdscli request GetAllSlammers @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllSlammers
ServiceType=NAS
```

See the GetAllSlammersResponse.txt file for example response data.

SEE ALSO

GetSlammerDetails on p. 295

Ranges for Option Values on p. 57

NAME **GetAllSnapLUNHierarchies**

DESCRIPTION Displays a list of all Snap LUN hierarchies.

SYNTAX

```
pdscli {request | submit} GetAllSnapLUNHierarchies
```

OPTIONS None

EXAMPLES Run the following command to display a list of all snap LUN hierarchies.

```
pdscli request GetAllSnapLUNHierarchies @input
```

See the GetAllSnapLUNHierarchiesResponse.txt file for example response data.

SEE ALSO

DeleteSnapLUN on p. 191

GetAllSnapLUNs on p. 233

GetSnapLUNDetails on p. 297

ModifySnapLUN on p. 381

PerformLUNDuplication on p. 447

PerformBackgroundLUNDuplication on p. 402

Ranges for Option Values on p. 57

NAME **GetAllSnapLUNs**

DESCRIPTION Displays a list of all Snap LUNs associated with a specified volume group.

SYNTAX

```
pdscli {request | submit} GetAllSnapLUNs
   Source=GUID-or-FQN-of-Snap LUN
```

OPTIONS **Source [optional]**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Snap LUN for which you want to display LUNs.

EXAMPLES Run the following command to list the Snap LUNs that are associated with EngLUN.

```
pdscli request GetAllSnapLUNs @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllSnapLUNs
Source=/EngLUN
```

See the GetAllLSnapUNsResponse.txt file for example response data.

SEE ALSO DeleteSnapLUN on p. 191

GetAllSnapLUNHierarchies on p. 232

GetSnapLUNDetails on p. 297

ModifySnapLUN on p. 381

PerformLUNDuplication on p. 447

PerformBackgroundLUNDuplication on p. 402

Ranges for Option Values on p. 57

NAME **GetAllTapeRobots**

DESCRIPTION Displays a list of tape devices that the Pillar Axiom storage system has discovered.

SYNTAX

```
pdscli {request | submit} GetAllTapeRobots
```

OPTIONS None.

EXAMPLES Run the following command to display a list of tape devices that are attached to your system to back up data that is stored on the system.

```
pdscli request GetAllTapeRobots
```

See the GetAllTapeRobotsResponse.txt file for example response data.

SEE ALSO PerformTapeRobotDiscovery on p. 504

Ranges for Option Values on p. 57

NAME **GetAllTasks**

DESCRIPTION Displays a list of tasks and the status of each.

SYNTAX

```
pdscli {request | submit} GetAllTasks
```

OPTIONS None.

EXAMPLES You want to know if any tasks show a status of “InProgress.” Run the following command to display a list of tasks.

```
pdscli request GetAllTasks
```

See the GetAllTasksResponse.txt file for example response data.

SEE ALSO GetTaskDetails on p. 304

GetAllJobs on p. 220

Ranges for Option Values on p. 57

NAME **GetAllUpdateComponents**

DESCRIPTION Displays a list of firmware and software modules that are installed in the system that can receive software updates.

SYNTAX

```
pdscli {request | submit} GetAllUpdateComponents
```

OPTIONS None.

EXAMPLES Run the following command to display a list of installed components in your Pillar Axiom storage system that can be updated.

```
pdscli request GetAllUpdateComponents
```

See the GetAllUpdateComponentsResponse.txt file for example response data.

SEE ALSO GetUpdatePackageDetails on p. 305

Ranges for Option Values on p. 57

NAME **GetAllUpdatePackages**

DESCRIPTION Displays a list of update packages that have been staged on the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetAllUpdatePackages
```

OPTIONS None.

EXAMPLES Run the following command to display a list of update packages that are staged on your system.

```
pdscli request GetAllUpdatePackages
```

See the GetAllUpdatePackagesResponse.txt file for example response data.

SEE ALSO GetUpdatePackageDetails on p. 305

Ranges for Option Values on p. 57

NAME **GetAllVerifyOperations**

DESCRIPTION The system automatically generates system verifications. This command displays a list of the system verifications.

SYNTAX

```
pdscli {request | submit}GetAllVerifyOperations
```

OPTIONS None.

EXAMPLES Run the following command to display a list of system verifications.

```
pdscli request GetAllVerifyOperations @input
```

See the GetAllVerifyOperationsResponse.txt file for example response data.

SEE ALSO GetAllTasks on p. 235

GetTaskDetails on p. 304

PerformCancelTask on p. 404

Ranges for Option Values on p. 57

NAME **GetAllVirtualInterfaces**

DESCRIPTION Displays a list of virtual network interfaces that are defined for the specified File Server.

SYNTAX

```
pdscli {request | submit} GetAllVirtualInterfaces
   FileServer=GUID-or-FQN-of-File-Server
```

OPTIONS **FileServer**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the File Server for which you want to display a list of virtual network interfaces.

EXAMPLES Run the following command to display a list of virtual interfaces that are defined for the Company-XYZ-FS1 File Server.

```
pdscli request GetAllVirtualInterfaces @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllVirtualInterfaces
FileServer=/Company-XYZ-FS1
```

See the GetAllVirtualInterfacesResponse.txt file for example response data.

SEE ALSO GetAllFileServers on p. 213

GetVirtualInterfaceDetails on p. 307

Ranges for Option Values on p. 57

NAME

GetAllVolumeGroups

DESCRIPTION

Displays a list of the volume groups that are used to organize logical volumes (filesystems and LUNs).

SYNTAX

```
pdscli {request | submit} GetAllVolumeGroups
   VolumeGroup=GUID-or-FQN-of-parent-volume-group
```

OPTIONS

VolumeGroup

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the next-highest volume group in the hierarchy. If volume groups are not nested, or if the next-highest level is the System itself, enter the GUID or FQN of the System.

EXAMPLES

Run the following command to display a list of volume groups that are defined on your system. To do so, enter the FQN of the System.

```
pdscli request GetAllVolumeGroups
```

where input.req contains the following name/value pairs:

```
Request=GetAllVolumeGroups
VolumeGroup=/Company-XYZ-data
```

See the GetAllVolumeGroupsResponse.txt file for example response data.

SEE ALSO

GetVolumeGroupDetails on p. 308

Ranges for Option Values on p. 57

NAME **GetBackupRunStatistics**

DESCRIPTION Displays performance statistics for the specified backups.

SYNTAX

```
pdscli {request | submit} GetBackupRunStatistics
   BackupRun=GUID-or-FQN-of-backup
```

OPTIONS

BackupRun

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all backups for which you want to display performance statistics.

EXAMPLES Run the following command to display performance statistics for the last backup that was performed on the engineering filesystems.

```
pdscli request GetBackupRunStatistics @input
```

where input.req contains the following name/value pairs:

```
Request=GetBackupRunStatistics
BackupRun=IDA0479DC2-38EF-9828-8902-
   E0FC03743270
```

See the GetBackupRunStatisticsResponse.txt file for example response data.

SEE ALSO GetAllBackupRuns on p. 203

Ranges for Option Values on p. 57

NAME **GetBrickDetails**

DESCRIPTION Displays information about the specified Bricks.

SYNTAX

```
pdscli {request | submit} GetBrickDetails
  Brick=GUID-or-FQN-of-Brick
```

OPTIONS **Brick**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all Bricks for which you want to display information.

EXAMPLES Run the following command to display information about the three Bricks that are configured on your Pillar Axiom storage system.

```
pdscli request GetBrickDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetBrickDetails
Brick=/Company-XYZ-data/Brick1
Brick=/Company-XYZ-data/Brick2
Brick=/Company-XYZ-data/Brick3
```

See the GetBrickDetailsResponse.txt file for example response data.

SEE ALSO GetAllBricks on p. 204

Ranges for Option Values on p. 57

NAME **GetCallHomeMatrixDetails**

DESCRIPTION Displays Call-Home Matrix details that have been sent to the Pillar World Wide Customer Support Center.

The Call-Home feature is defined as:

A Pillar Axiom feature that, when enabled, allows the system to notify the Pillar World Wide Customer Support Center of critical issues specific to the Pillar Axiom storage system. No customer data is transmitted. The Call-Home feature transfers files over the Internet using one of the following user-selected methods:

- SCP: Uses secure copy (SCP) with 1024-bit encryption and secure keys.
- HTTPS: Sends files either directly to the Pillar World Wide Customer Support Center or through a proxy server for security purposes. Can also be used when the Pillar Axiom storage system does not have direct access to the Internet.

SYNTAX

```
pdscli {request | submit}
  GetCallHomeMatrixDetails
```

OPTIONS None.

EXAMPLES Run the following command to display results of the latest details.

```
pdscli
```

See the GetCallHomeMatrixDetailsResponse.txt file for example response data.

NAME **GetCIFSShareDetails**

DESCRIPTION Displays information about the specified Common Internet File System (CIFS) shares.

SYNTAX

```
pdscli {request | submit} GetCIFSShareDetails
   CIFSShare=GUID-or-FQN-of-CIFS-share
```

OPTIONS **CIFSShare**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all CIFS shares for which you want to display information.

EXAMPLES Run the following command to display information about a CIFS share that is associated with the EngAtl filesystem.

```
pdscli request GetCIFSShareDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetCIFSShareDetails
CIFSShare=/Company-XYZ-FS1/EngAtl/src
```

See the GetCIFSShareDetailsResponse.txt file for example response data.

SEE ALSO GetAllCIFSShares on p. 206

Ranges for Option Values on p. 57

NAME **GetCloneFSDetails**

DESCRIPTION Displays all details of a specified Clone FS.

A Clone FS is a point-in-time, writeable snapshot of a filesystem, which can be used later to revert the filesystem. This snapshot retains the same QoS parameters as the source filesystem. It consumes storage capacity from the Clone FS storage space created for the source filesystem. A Clone FS can be scheduled to occur at any time. This provides a convenient method to branch from the source data without the need to do a full block-level copy.

SYNTAX

```
pdscli {request | submit}
   GetCloneFSDetails
   CloneFSFQN=FQN-of-clonefs
```

OPTIONS **CloneFSFQN**

Identifies a specific Pillar Axiom object.

- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the FQN of the filesystem with which this share is associated. If there is more than one file server, use the FQN that includes the file server and filesystem.

EXAMPLES Run the following command to view details for a Clone FS named Company-XYZ-FS1.

```
pdscli request GetCloneFSDetails @input
```

where input.req contains the following name/value pairs:

```
Request=etCloneFSDetails
CloneFSFQN==/Company-XYZ-FS1
```

See the GetCloneFSDetailsResponse.txt file for example response data.

SEE ALSO CreateCloneFS on p. 97

DeleteCloneFS on p. 172

GetAllCloneFSHierarchies on p. 207

GetAllCloneFSs on p. 208

GetAllSnapLUNs on p. 233

ModifyCloneFS on p. 317

NAME **GetCurrentDateTimeDetails**

DESCRIPTION Displays information about the current date and time on the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetCurrentDateTimeDetails
```

OPTIONS None.

EXAMPLES Runs the following command to display information about the time on your system.

```
pdscli request GetCurrentDateTimeDetails
```

See the GetCurrentDateTimeDetailsResponse.txt file for example response data.

SEE ALSO ModifyTimeConfig on p. 386

Ranges for Option Values on p. 57

NAME **GetCurrentTimeZoneDetails**

DESCRIPTION Displays information about the current time zone that is defined for the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetCurrentTimeZoneDetails
```

OPTIONS None.

EXAMPLES Run the following command to display information about the time zone that is defined for your system.

```
pdscli request GetCurrentTimeZoneDetails
```

See the GetCurrentTimeZoneDetailsResponse.txt file for example response data.

SEE ALSO GetCurrentDateTimeDetails on p. 247

Ranges for Option Values on p. 57

NAME **GetDataConsistencyDetails**

DESCRIPTION Displays results from the most recent run of the PerformVerifyDataConsistency request.

SYNTAX

```
pdscli {request | submit} GetDataConsistencyDetails
```

OPTIONS None.

EXAMPLES Run the following command to display results of the latest data consistency verification.

```
pdscli request GetDataConsistencyDetails
```

See the GetDataConsistencyDetailsResponse.txt file for example response data.

SEE ALSO PerformVerifyDataConsistency on p. 515

Ranges for Option Values on p. 57

NAME **GetExternalUPSDeviceDetails**

DESCRIPTION Displays information about a specific external uninterruptable power supply device.

SYNTAX

```
pdscli {request | submit} GetExternalUPSDeviceDetails
   ups=GUID-or-FQN-of-UPS-device
```

OPTIONS

ups

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the UPS device to modify.

EXAMPLES Run the following command to display information about the UPS1 device.

```
pdscli request GetExternalUPSDeviceDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetExternalUPSDeviceDetails
ups=UPS1
```

See the GetExternalUPSDeviceDetailsResponse.txt file for example response data.

SEE ALSO GetAllExternalUPSDevices on p. 212

Ranges for Option Values on p. 57

NAME **GetFCPortStatistics Details**

DESCRIPTION Displays Fibre Channel (FC) port statistics.

SYNTAX

```
pdscli {request | submit} GetFCPortStatisticsDetails
  NetworkInterfaceIOPort.Slammer=
    GUID-or-FQN-of-NAS Slammer
  NetworkInterfaceIOPort.ControlUnitNumber=
    {0 | 1}
  NetworkInterfaceIOPort.EthernetPortName=
    {Port0 | Port1}
```

OPTIONS

NetworkInterfaceIOPort.Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer.

NetworkInterfacePort.ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

NetworkInterfacePort.EthernetPortName

Identifies the port on the Slammer that is assigned to the File Server's virtual network interface. Select a gigabit Ethernet network port on on one of the data-path Slammers control unit (CU) .

- If Link Aggregation is enabled, select port 0 (zero). You cannot create virtual interfaces on alternate port 1 when Link Aggregation is enabled.
- If Link Aggregation is disabled, select either port.

EXAMPLES Run the following command to display FCPort statistics for Slammer1.

```
pdscli request GetFCPortStatisticsDetails @input
```

where input.req contains the following name/value pairs:

```
Request = GetFCPortStatisticsDetails
NetworkInterfaceIOPort.Slammer = Slammer1
NetworkInterfaceIOPort.ControlUnitNumber = 1
NetworkInterfaceIOPort.EthernetPortName = PORT0
```

See the GetFCPortStatisticsDetailsResponse.txt file for example response data.

SEE ALSO

GetFileServerStatistics (p. 254)

GetFileSystemStatistics (p. 260)

GetiSCSIPortStatisticsDetails (p. 266)

GetLUNStatistics (p. 272)

NAME **GetFileServerDetails**

DESCRIPTION Displays information about the specified File Servers.

SYNTAX

```
pdscli {request | submit} GetFileServerDetails
   FileServer=GUID-or-FQN-of-File-Server
```

OPTIONS **FileServerID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all File Servers for which you want to display information.

EXAMPLES Run the following command to display information about the Company-XYZ-FS1 File Server.

```
pdscli request GetFileServerDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetFileServerDetails
FileServer=/Company-XYZ-FS1
```

See the GetFileServerDetailsResponse.txt file for example response data.

SEE ALSO GetAllFileServers on p. 213

GetFileServerStatistics on p. 254

Ranges for Option Values on p. 57

NAME **GetFileServerStatistics**

DESCRIPTION Displays performance statistics for the specified File Servers.

SYNTAX

```
pdscli {request | submit} GetFileServerStatistics
   ID=GUID-or-FQN-of-File-Server
```

OPTIONS

ID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all File Servers for which you want to display performance statistics.

EXAMPLES Run the following command to display performance statistics for the Company-XYZ-FS1 File Server.

```
pdscli request GetFileServerStatistics @input
```

where input.req contains the following name/value pairs:

```
Request=GetFileServerStatistics
ID=/Company-XYZ-FS1
```

See the GetFileServerStatisticsResponse.txt file for example response data.

SEE ALSO

GetAllFileServers on p. 213

GetFileServerDetails on p. 253

Ranges for Option Values on p. 57

NAME **GetFileSystemDetails**

DESCRIPTION Displays information about the specified filesystems.

SYNTAX

```
pdscli {request | submit} GetFileSystemDetails
   ID=GUID-or-FQN-of-filesystem
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all filesystems for which you want to display information.

EXAMPLES Run the following command to display information about the EngAtl filesystem.

```
pdscli request GetFileSystemDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetFileSystemDetails
ID=/Company-XYZ-FS1/EngAtl
```

See the GetFileSystemDetailsResponse.txt file for example response data.

SEE ALSO GetAllFileSystems on p. 214

GetFileSystemStatistics on p. 260

Ranges for Option Values on p. 57

NAME **GetFileSystemSnapshotDetails**

DESCRIPTION Displays information about the specified Snap FSs.

SYNTAX

```
pdscli {request | submit}
   GetFileSystemSnapshotDetails
      ID=GUID-or-FQN-of-Snap-FS
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all Snap FSs for which you want to display information.

EXAMPLES Run the following command to display information about a Snap FS that was created from the EngAtl filesystem.

```
pdscli request GetFileSystemSnapshotDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetFileSystemSnapshotDetails
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GetFileSystemSnapshotDetailsResponse.txt file for example response data.

SEE ALSO CreateFileSystemSnapshotSchedule on p. 124

DeleteFileSystemSnapshot on p. 177

DeleteFileSystemSnapshotSchedule on p. 179

GetAllFileSystemSnapshots on p. 216

GetAllFileSystemSnapshotSchedules on p. 218

GetFileSystemSnapshotScheduleDetails on p. 258

ModifyFileSystemSnapshot on p. 340

PerformFileSystemDuplication on p. 425

PerformBackgroundFileSystemDuplication on p. 400

PerformRestoreFileSystemFromSnapshot on p. 485

PerformRebuildFileSystemRedundancy on p. 474

Ranges for Option Values on p. 57

NAME **GetFileSystemSnapshotScheduleDetails**

DESCRIPTION Displays information about the specified Snap FS schedules.

SYNTAX

```
pdscli {request | submit}
  GetFileSystemSnapshotScheduleDetails
    ID=GUID-or-FQN-of-Snap-FS-schedule
```

OPTIONS

ID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all Snap FS schedules for which you want to display information.

EXAMPLES Run the following command to display information about a schedule that creates Snap FSs of the EngAtl filesystem.

```
pdscli request GetFileSystemSnapshotScheduleDetails
  @input
```

where input.req contains the following name/value pairs:

```
Request=GetFileSystemSnapshotScheduleDetails
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GetFileSystemSnapshotScheduleDetails Response.txt file for example response data.

SEE ALSO

CreateFileSystemSnapshotSchedule on p. 124

DeleteFileSystemSnapshot on p. 177

DeleteFileSystemSnapshotSchedule on p. 179

GetAllFileSystemSnapshots on p. 216

GetAllFileSystemSnapshotSchedules on p. 218

GetFileSystemSnapshotDetails on p. 256

ModifyFileSystemSnapshot on p. 340

PerformFileSystemDuplication on p. 425

PerformBackgroundFileSystemDuplication on p. 400

PerformRestoreFileSystemFromSnapshot on p. 485

PerformRebuildFileSystemRedundancy on p. 474

Ranges for Option Values on p. 57

NAME **GetFileSystemStatistics**

DESCRIPTION Displays performance statistics for the specified filesystems.

SYNTAX

```
pdscli {request | submit} GetFileSystemStatistics
   ID=GUID-or-FQN-of-filesystem
```

OPTIONS

ID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all filesystems for which you want to display performance statistics.

EXAMPLES Run the following command to display performance statistics for the EngAtl filesystem.

```
pdscli request GetFileSystemStatistics @input
```

where input.req contains the following name/value pairs:

```
Request=GetFileSystemStatistics
ID=/Company-XYZ-FS1/EngAtl
```

See the GetFileSystemStatisticsResponse.txt file for example response data.

SEE ALSO GetAllFileSystems on p. 214

GetFileSystemDetails on p. 255

Ranges for Option Values on p. 57

NAME **GetGUISSLConfigDetails**

DESCRIPTION Displays information about whether the Pillar Axiom Storage Services Manager graphical user interface (GUI) is configured to use secure sockets layer (SSL) certificates. If SSL certificates are used, certificate information is displayed.

SYNTAX

```
pdscli {request | submit} GetGUISSLConfigDetails
```

OPTIONS None.

EXAMPLES Run the following command to display information about the SSL certificate that is configured on your system.

```
pdscli request GetGUISSLConfigDetails
```

See the GetGUISSLConfigDetailsResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **GetHBAPortConnections**

DESCRIPTION Lists information for a specified HBA port.

SYNTAX

```
pdscli {request | submit} GetHBAPortConnections
   HBAPortWWN=World-wide-number
```

OPTIONS **HBAPortWWN**

Identifies the address, or world wide name (WWN), of the specified host bus adapter (HBA).

EXAMPLES Run the following command to list information for an HBA port.

```
pdscli request GetHBAPortConnections @input
```

where input.req contains the following name/value pairs:

```
Request=GetHBAPortConnections
HBAPortWWN=21:00:00:e0:8b:11:b5:4a
```

See the GetHBAPortConnectionsResponse.txt file for example response data.

SEE ALSO GetSANConfigDetails on p. 284

GetHostMappingConfigDetails on p. 265

GetAllSANHosts on p. 227

ModifySANHost on p. 377

PerformAssociateInitiatorsToHost on p. 398

DeleteSANHost on p. 189

Ranges for Option Values on p. 57

NAME **GetHistoricalFileSystemStatistics**

DESCRIPTION Displays performance statistics over the specified interval for the specified filesystems.

SYNTAX

```
pdscli {request | submit}
  GetHistoricalFileSystemStatistics
    FileSystem=GUID-or-FQN-of-filesystem
    Interval={Always | Hourly | EightHourly |
      Daily | Weekly | Monthly | Quarterly |
      HalfYearly | Yearly}
    TimeFilter.AfterTimestamp=
      YYYY-MM-DDTHH:mm:SS.xx+-HH:mm
    TimeFilter.BeforeTimestamp=
      YYYY-MM-DDTHH:mm:SS.xx+-HH:mm
```

OPTIONS

FileSystem

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all filesystems for which you want to display historical performance statistics.

Interval

Identifies the intervals at which the Pillar Axiom system calculates historical statistics. Choose from:

- Always
- Hourly
- EightHourly
- Daily
- Weekly
- Monthly
- Quarterly
- HalfYearly
- Yearly

TimeFilter.AfterTimestamp AND TimeFilter.BeforeTimestamp

Identifies the time period for data collection. Use the following format for the entry:

YYYY-MM-DDTHH:mm:SS.xx+-HH:mm

where:

- YYYY-MM-DD designates a four-digit year, two-digit month, and two-digit day for the date.
- T is a separator that designates the start of the time portion of the string.
- HH:mm:SS designates hours, minutes, and seconds in values for a 24-hour clock.
- xx designates a fraction of a second, to two decimal places.
- +-HH:mm designates the time zone as an offset from Coordinated Universal Time (UTC) in hours and minutes. Include the + or - prefix as appropriate for your time zone in relation to UTC.

EXAMPLES

Run the following command to display daily performance statistics for the EngAtl filesystem in the month of June.

```
pdscli request GetHistoricalFileSystemStatistics
   @input
```

where input.req contains the following name/value pairs:

```
Request=GetHistoricalFileSystemStatistics
FileSystem=/Company-XYZ-FS1/EngAtl
Interval=Daily
TimeFilter.AfterTimestamp=
   2004-06-01T00:01:00.00-08:00
TimeFilter.BeforeTimestamp=
   2004-06-30T00:59:59.59-08:00
```

See the GetHistoricalFileSystemStatisticsResponse.txt file for example response data.

SEE ALSO

GetFileSystemStatistics on p. 260

GetFileSystemDetails on p. 255

Ranges for Option Values on p. 57

NAME

GetHostMappingConfigDetails

DESCRIPTION

Displays storage assignment and configuration information for the LUNs that are mapped to a specific SAN host.

SYNTAX

```
pdscli {request | submit} GetHostMappingConfigDetails
   SANHostName=Host-name
```

OPTIONS

SANHostName

Identifies the name assigned to a specific SAN host.

EXAMPLES

Run the following command to display information about the SANTest_02 host.

```
pdscli request GetHostMappingConfigDetails@input
```

where input.req contains the following name/value pairs:

```
Request=GetHostMappingConfigDetails
SANHostName=SANTest_02
```

See the GetHostMappingConfigDetailsResponse.txt file for example response data.

SEE ALSO

GetSANConfigDetails on p. 284

GetAllSANHosts on p. 227

ModifySANHost on p. 377

GetHBAPortConnections on p. 262

PerformAssociateInitiatorsToHost on p. 398

DeleteSANHost on p. 189

Ranges for Option Values on p. 57

NAME **GetiSCSIPortStatisticsDetails**

DESCRIPTION Displays iSCSI port statistics.

SYNTAX

```
pdscli {request | submit} GetiSCSIPortStatistics
  NetworkInterfaceIOPort.Slammer=
    GUID-or-FQN-of-SAN Slammer
  NetworkInterfaceIOPort.ControlUnitNumber=
    {0 | 1}
  NetworkInterfaceIOPort.EthernetPortName=
    {Port0 | Port1}
```

OPTIONS

NetworkInterfaceIOPort.Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the SAN Slammer.

NetworkInterfacePort.ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

NetworkInterfacePort.EthernetPortName

Identifies the port on the Slammer that is assigned to the File Server's virtual network interface. Select a gigabit Ethernet network port on on one of the data-path Slammers control unit (CU) .

- If Link Aggregation is enabled, select port 0 (zero). You cannot create virtual interfaces on alternate port 1 when Link Aggregation is enabled.
- If Link Aggregation is disabled, select either port.

EXAMPLES Run the following command to display iSCSI port statistics for Slammer1.

```
pdscli request GetiSCSIPortStatisticsDetails @input
```

where input.req contains the following name/value pairs:

```
Request = GetiSCSIPortStatisticsDetails
NetworkInterfaceIOPort.Slammer = Slammer1
NetworkInterfaceIOPort.ControlUnitNumber = 1
NetworkInterfaceIOPort.EthernetPortName = PORT0
```

See the GetiSCSIPortStatisticsDetailsResponse.txt file for example response data.

SEE ALSO

GetFCPortStatistics Details (p. 251)

GetSANPortDetails (p. 286)

GetSANProtocolStatistics (p. 288)

GetSlammerDetails (p. 295)

ModifyiSCSIPortDetails (p. 343)

NAME **GetJobDetails**

DESCRIPTION Displays information about the specified jobs. Jobs are scheduled tasks.

SYNTAX

```
pdscli {request | submit} GetJobDetails
   Job=GUID-or-FQN-of-job
```

OPTIONS **Job**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of all jobs for which you want to display information.

EXAMPLES Run the following command to display information about the Snap FSs for the EngAtl filesystem.

```
pdscli request GetJobDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetJobDetails
Job[6]=IDA0479DC2-38EF-9828-8902-E0FC03743270
Job[7]=IDf1234567-89ab-cde0-0123-456789eeeee0
```

See the GetJobDetailsResponse.txt file for example response data.

SEE ALSO

GetAllJobs on p. 220

GetAllSchedules on p. 229

GetAllTasks on p. 235

Ranges for Option Values on p. 57

NAME **GetLUNDetails**

DESCRIPTION Displays information about a specified LUN. A LUN is defined as:

> A logical volume within a storage area network (SAN). Administrators assign storage resources and Quality of Service (QoS) attributes to each logical unit (LUN).

SYNTAX

```
pdscli {request | submit} GetLUNDetails
   ID=GUID-or-FQN-of-LUN
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all LUNs for which you want to display information.

EXAMPLES Run the following command to display information about EngLUN.

```
pdscli request GetLUNDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetLUNDetails
ID=/EngLUN
```

See the GetLUNDetailsResponse.txt file for example response data.

SEE ALSO CreateLUN on p. 135

DeleteLUN on p. 182

GetAllLUNs on p. 221

GetHostMappingConfigDetails on p. 265

GetLUNMappingConfigDetails on p. 270

ModifyLUN on p. 351

PerformConfigureLUNMapping on p. 412

PerformRebuildLUNRedundancy on p. 475

Ranges for Option Values on p. 57

NAME **GetLUNMappingConfigDetails**

DESCRIPTION Displays storage assignment and configuration information for a specified LUN(s) or SnapLUN(s) that are mapped.

SYNTAX

```
pdscli {request | submit} GetLUNMappingConfigDetails
   MappedLUNID=GUID-of-LUN
   MappedLUNFQN=FQN-of-LUN
```

OPTIONS **MappedLUN**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Mapped LUN or SnapLUN for which you want to display information.

EXAMPLES Run the following command to display information about EngLUN.

```
pdscli request GetLUNMappingConfigDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetLUNMappingConfigDetails
LUN=/EngLUN
```

See the GetLUNMappingConfigDetailsResponse.txt file for example response data.

SEE ALSO CreateLUN on p. 135

DeleteLUN on p. 182

GetAllLUNs on p. 221

GetHostMappingConfigDetails on p. 265

GetLUNDetails on p. 269

ModifyLUN on p. 351

PerformConfigureLUNMapping on p. 412

PerformRebuildLUNRedundancy on p. 475

Ranges for Option Values on p. 57

NAME **GetLUNStatistics**

DESCRIPTION Display statistics for a specified LUN.

SYNTAX

```
pdscli {request | submit} GetLUNStatistics
   ID=GUID-or-FQN-of-LUN
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the LUN for which you want to display information.

EXAMPLES Run the following command to display information about EngLUN.

```
pdscli request GetLUNStatistics @input
```

where input.req contains the following name/value pairs:

```
Request=GetLUNStatistics
ID=/EngLUN
```

See the GetLUNStatisticsResponse.txt file for example response data.

SEE ALSO GetFileSystemStatistics on p. 260

Ranges for Option Values on p. 57

NAME **GetManagementConfigDetails**

DESCRIPTION Displays information about the global settings that are defined for the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetManagementConfigDetails
```

OPTIONS None.

EXAMPLES Run the following command to display information about the global settings in your system.

```
pdscli request GetManagementConfigDetails
```

See the GetManagementConfigDetailsResponse.txt file for example response data.

SEE ALSO GetGUISSLConfigDetails (p. 261)

GetHostMappingConfigDetails (p. 265)

GetLUNMappingConfigDetails (p. 270)

GetNASConfigDetails (p. 274)

GetNDMPConfigDetails (p. 275)

GetNTPServerConfigDetails (p. 279)

GetSANConfigDetails (p. 284)

GetSANPortDetails (p. 286)

GetSANSlammerPortDetails (p. 290)

GetSNMPConfigDetails (p. 298)

NAME

GetNASConfigDetails

DESCRIPTION

Displays information about the NAS storage configuration in the Pillar Axiom storage system. Details include the number of filesystems, capacity usage, and capacity that is reserved for overcommitted filesystems.

SYNTAX

```
pdscli {request | submit} GetNASConfigDetails
```

OPTIONS

None.

EXAMPLES

Run the following command to display information about the NAS configuration in your system.

```
pdscli request GetNASConfigDetails
```

See the GetNASConfigDetailsResponse.txt file for example response data.

SEE ALSO

GetStorageConfig on p. 300

GetAllFileSystems on p. 214

GetFileSystemDetails on p. 255

Ranges for Option Values on p. 57

NAME

GetNDMPConfigDetails

DESCRIPTION

Displays information about the Network Data Management Protocol (NDMP) configuration settings.

SYNTAX

```
pdscli {request | submit} GetNDMPConfigDetails
```

OPTIONS

None.

EXAMPLES

Run the following command to display information about the NDMP configuration in your system.

```
pdscli request GetNDMPConfigDetails
```

See the GetNDMPConfigDetailsResponse.txt file for example response data.

SEE ALSO

GetProductDetails on p. 281

Ranges for Option Values on p. 57

NAME **GetNFSExportDetails**

DESCRIPTION Displays information about the specified Network File System (NFS) exports.

SYNTAX

```
pdscli {request | submit} GetNFSExportDetails
  NFSExport=GUID-or-FQN-of-NFS-export
```

OPTIONS **NFSExport**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all NFS exports for which you want to display information.

EXAMPLES Run the following command to display information about an export for the EngAtl filesystem.

```
pdscli request GetNFSExportDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetNFSExportDetails
NFSExport=IDA0479DC2-38EF-9828-8902-
  E0FC03743270
```

See the GetNFSExportDetailsResponse.txt file for example response data.

SEE ALSO GetAllNFSExports on p. 222

Ranges for Option Values on p. 57

NAME **GetNISUploadedFilesDetails**

DESCRIPTION Displays information about the password, group, and netgroup files that are defined as Network Information Service (NIS) alternatives for the specified File Server.

SYNTAX

```
pdscli {request | submit}
  GetNISUploadedFilesDetails
    FileServer=GUID-or-FQN-of-File-Server
```

OPTIONS **FileServer**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all File Servers for which you want to display information.

EXAMPLES Run the following command to display details about the NIS-alternative files that are associated with the Company-XYZ-FS1 File Server.

```
pdscli request GetNISUploadedFilesDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetNISUploadedFilesDetails
FileServer=/Company-XYZ-FS1
```

See the GetNISUploadedFilesDetailsResponse.txt file for example response data.

SEE ALSO PerformNISFileUpload on p. 456

Ranges for Option Values on p. 57

NAME **GetNotificationDefinitionDetails**

DESCRIPTION Displays information about the specified notifications.

SYNTAX

```
pdscli {request | submit}
  GetNotificationDefinitionDetails
    NotificationDefinition=GUID-or-FQN-of-
      notification-definition
```

OPTIONS

NotificationDefinition

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all notifications for which you want to display information.

EXAMPLES Run the following command to display information about the alert that is emailed to designated recipients when critical events occur in your system.

```
pdscli request GetNotificationDefinitionDetails
  @input
```

where input.req contains the following name/value pairs:

```
Request=GetNotificationDefinitionDetails
NotificationDefinition=IDA0479DC2-38EF-
  9828-8902-E0FC03743270
```

See the GetNotificationDefinitionDetailsResponse.txt file for example response data.

SEE ALSO GetAllNotificationDefinitions on p. 223

Ranges for Option Values on p. 57

NAME **GetNTPServerConfigDetails**

DESCRIPTION Displays information about the Network Time Protocol (NTP) configuration settings.

SYNTAX

```
pdscli {request | submit} GetNTPServerConfigDetails
```

OPTIONS None.

EXAMPLES Run the following command to display information about the NTP servers with which the clocks in your system are synchronized.

```
pdscli request GetNTPServerConfigDetails
```

See the GetNTPServerConfigDetailsResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **GetPilotDetails**

DESCRIPTION Displays information about the Pillar Axiom Pilot.

SYNTAX

```
pdscli {request | submit} GetPilotDetails
```

OPTIONS None.

EXAMPLES Run the following command to display information about the Pilot hardware component in your system.

```
pdscli request GetPilotDetails
```

See the GetPilotDetailsResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **GetProductDetails**

DESCRIPTION Displays information about the Pillar Axiom storage system. This request was designed for NDMP clients (DMAs) that back up and restore data on the system. The request results can be useful in other scenarios, too.

SYNTAX

```
pdscli {request | submit} GetProductDetails
```

OPTIONS None.

EXAMPLES Run the following command to display information about your system.

```
pdscli request GetProductDetails
```

See the GetProductDetailsResponse.txt file for example response data.

SEE ALSO GetNDMPConfigDetails on p. 275

Ranges for Option Values on p. 57

NAME **GetQuotaDetails**

DESCRIPTION Displays information about the specified quotas that are defined for NAS filesystems.

SYNTAX

```
pdscli {request | submit} GetQuotaDetails
   ID=GUID-or-FQN-of-quota
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all filesystem quotas for which you want to display information.

EXAMPLES Run the following command to display information about a quota that is defined for the engineering organization's filesystem, EngAtl.

```
pdscli request GetQuotaDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetQuotaDetails
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GetQuotaDetailsResponse.txt file for example response data.

SEE ALSO GetAllQuotas on p. 224

Ranges for Option Values on p. 57

NAME **GetRouteDetails**

DESCRIPTION Displays information about the specified secondary routes that are defined for a File Server. The primary route is an attribute of the File Server itself.

SYNTAX

```
pdscli {request | submit} GetRouteDetails
   Route=GUID-or-FQN-of-network-route
```

OPTIONS **Route**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all routes for which you want to display information.

EXAMPLES Run the following command to display information about a route that you defined for `Company`-XYZ-FS1 File Server.

```
pdscli request GetRouteDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetRouteDetails
Route=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GetRouteDetailsResponse.txt file for example response data.

SEE ALSO GetAllRoutes on p. 226

GetAllFileServers on p. 213

Ranges for Option Values on p. 57

NAME **GetSANConfigDetails**

DESCRIPTION Displays information about the system-wide SAN objects that are configured on the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetSANConfigDetails
```

OPTIONS None.

EXAMPLES Run the following command to display information about the SAN objects that are configured on your system.

```
pdscli request GetSANConfigDetails
```

See the GetSANConfigDetailsResponse.txt file for example response data.

SEE ALSO

GetHostMappingConfigDetails on p. 265

GetAllSANHosts on p. 227

ModifySANHost on p. 377

GetHBAPortConnections on p. 262

PerformAssociateInitiatorsToHost on p. 398

DeleteSANHost on p. 189

Ranges for Option Values on p. 57

NAME **GetSANHostDetails**

DESCRIPTION Displays information for a specific SAN host.

SYNTAX

```
pdscli {request | submit} GetSANHostDetails
   SANHostName=Host-name
```

OPTIONS

SANHostName

Identifies the name assigned to a specific SAN host.

EXAMPLES Run the following command to display information about the SANTest_02 host.

```
pdscli request GetSANHostDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetSANHostDetails
SANHostName=SANTest_02
```

See the GetSANHostDetailsResponse.txt file for example response data.

SEE ALSO GetHostMappingConfigDetails on p. 265

Ranges for Option Values on p. 57

NAME **GetSANPortDetails**

DESCRIPTION Displays detailed information about the SAN port.

SYNTAX

```
pdscli {request | submit} GetSANPortDetails
  SANPortIdentifier.Slammer=GUID-or-FQN
  SANPortIdentifier.ControlUnitNumber=
    {0 | 1}
  SANPortIdentifier.PortType=
    {FibreChannel | iSCSI}
  SANPortIdentifier.PortName=
    {Port1 | Port0}
```

OPTIONS

SANPortIdentifier.Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

SANPortIdentifier.ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

SANPortIdentifier.PortType

Identifies the SAN Slammer port type: iSCSI or Fibre Channel.

SANPortIdentifier.PortName

Identifies the Slammer port name: Port0 or Port1.

EXAMPLES Run the following command to display information about both Slammers that are configured on your system.

```
pdscli request GetSANPortDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetSANPortDetails
SANPortIdentifier.Slammer=/Company-XZY-data/Slammer1
SANPortIdentifier.ControlUnitNumber=1
SANPortIdentifier.PortType=FibreChannel
SANPortIdenifier.PortName=Port1
```

See the GetSANPortDetailsResponse.txt file for example response data.

See Also

GetAllSlammers on p. 231

GetFCPortStatistics Details on p. 251

GetiSCSIPortStatisticsDetails on p. 266

GetSANConfigDetails on p. 284

GetSANSlammerPortDetails on p. 290

GetSlammerDetails on p. 295

NAME **GetSANProtocolStatistics**

DESCRIPTION Displays the SAN statistics for a particular SAN network interface in the Pillar Axiom system.

SYNTAX

```
pdscli {request | submit} GetSANProtocolStatistics
   NetworkInterfaceIOPort.Slammer=
      GUID-or-FQN-of-NAS Slammer
   NetworkInterfaceIOPort.ControlUnitNumber=
      {0 | 1}
   NetworkInterfaceIOPort.EthernetPortName=
      {Port0 | Port1}
```

OPTIONS

NetworkInterfaceIOPort.Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer.

NetworkInterfacePort.ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs. The CUs are identified by 0 (the top CU) and 1 (the bottom CU).

NetworkInterfacePort.EthernetPortName

Identifies the port on the Slammer that is assigned to the File Server's virtual network interface. Select a gigabit Ethernet network port on on one of the data-path Slammers control unit (CU) .

- If Link Aggregation is enabled, select port 0 (zero). You cannot create virtual interfaces on alternate port 1 when Link Aggregation is enabled.
- If Link Aggregation is disabled, select either port.

EXAMPLES Run the following command to display SAN protocol statistics port 0 of control unit 1 on Slammer1.

```
pdscli request GetSANProtocolStatistics @input
```

where input.req contains the following name/value pairs:

```
Request = GetSANProtocolStatistics
```

```
NetworkInterfaceIOPort.Slammer = Slammer1
NetworkInterfaceIOPort.ControlUnitNumber = 1
NetworkInterfaceIOPort.EthernetPortName = PORT0
```

See the GetSANProtocolStatistics Response.txt file for example response data.

SEE ALSO GetSANConfigDetails on p. 284

Ranges for Option Values on p. 57

NAME **GetSANSlammerPortDetails**

DESCRIPTION Displays detailed information about the SAN Slammer port.

SYNTAX

```
pdscli {request | submit} GetSANSlammerPortDetails
     SlammerID=GUID-of-Slammer
     SlammerFQN=FQN-of-Slammer
```

OPTIONS

SlammerID

Enter the globally unique ID (GUID), which is a 38-character string that starts with “ID” and ends with 36 lower-case hex characters (a-f and 0-9) and hyphens.

SlammerFQN

Enter the fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name.

FQN format;

/Fully/Qualified/ObjectName

EXAMPLES Run the following command to display information about both Slammers that are configured on your system.

```
pdscli request GetSANSlammerPortDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetSANSlammerPortDetails
SlammerID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GetSANSlammerPortDetailsResponse.txt file for example response data.

See Also

GetiSCSIPortStatisticsDetails on p. 266

GetSANConfigDetails on p. 284

GetSANPortDetails on p. 286

GetSANSlammerPortDetails on p. 290

GetSlammerDetails on p. 295

NAME **GetScheduleDetails**

DESCRIPTION Displays information about the specified schedule.

SYNTAX

```
pdscli {request | submit} GetScheduleDetails
   Schedule=GUID-or-FQN-of-schedule
```

OPTIONS **Schedule**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all schedules for which you want to display information.

EXAMPLES Run the following command to display information about a schedule that creates a Snap FS of the EngAtl filesystem.

```
pdscli request GetScheduleDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetScheduleDetails
Schedule=IDA0479DC2-38EF-9828-8902-
   E0FC03743270
```

See the GetScheduleDetailsResponse.txt file for example response data.

SEE ALSO GetAllSchedules on p. 229

GetAllFileSystemSnapshotSchedules on p. 218

Ranges for Option Values on p. 57

NAME

GetSlammerCommandOutput

DESCRIPTION

Displays Slammer command output from a previously run PerformSlammerCommand request. The output allows you to view results of commands related to networking, primarily to help troubleshoot networking problems.

SYNTAX

```
pdscli {request | submit} GetSlammerCommandOutput
   CommandID=
```

OPTIONS

CommandID

Identifies the Slammer command output you wish to view.

EXAMPLES

Run the following command to display Slammer command output.

```
pdscli request GetSlammerCommandOutput @input
```

where input.req contains the following name/value pairs:

```
Request=GetSlammerCommandOutput
CommandID =IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the GetSlammerCommandOutputResponse.txt file for example response data.

SEE ALSO

GetAllSlammers on p. 231

GetSlammerDetails on p. 295

GetSlammerStatistics on p. 296

ModifySlammer on p. 380

PerformSlammerCommand (p. 496)

Ranges for Option Values on p. 57

NAME **GetSlammerControlUnitDiagnosticDetails**

DESCRIPTION Displays Slammer control unit diagnostic details. Use this command to troubleshoot Slammer hardware or view results of a power-on self test (POST). This request requires the installation of test devices and should never be initiated without the participation of the Pillar World Wide Customer Support Center.

Important! The system will shut down the control unit when you issue this command.

SYNTAX

```
pdscli {request | submit} GetSlammerControlUnitDiagnosticDetails
   Slammer=GUID-or-FQN-of-UPS-device
   ControlUnitNumber={0 | 1}
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs. The CUs are identified by 0 (the top CU) and 1 (the bottom CU).

EXAMPLES Run the following command to display Slammer command output.

```
pdscli request GetSlammerControlUnitDiagnosticDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetSlammerControlUnitDiagnostic
   Details
Slammer=/Company-XZY-data/Slammer1
ControlUnitNumber=1
```

See the GetSlammerControlUnitDiagnosticDetails
Response.txt file for example response data.

SEE ALSO GetAllSlammers on p. 231

GetSlammerDetails on p. 295

GetSlammerStatistics on p. 296

ModifySlammer on p. 380

Ranges for Option Values on p. 57

NAME **GetSlammerDetails**

DESCRIPTION Displays information about the specified Slammers.

SYNTAX

```
pdscli {request | submit} GetSlammerDetails
   Slammer=GUID-or-FQN-of-Slammer
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all Slammers for which you want to display information.

EXAMPLES Run the following command to display information about both Slammers that are configured on your system.

```
pdscli request GetSlammerDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetSlammerDetails
Slammer=/Company-XZY-data/Slammer1
Slammer=/Company-XZY-data/Slammer2
```

See the GetSlammerDetailsResponse.txt file for example response data.

SEE ALSO GetAllSlammers on p. 231

GetSlammerStatistics on p. 296

Ranges for Option Values on p. 57

NAME **GetSlammerStatistics**

DESCRIPTION Displays performance statistics for the specified Slammers.

SYNTAX

```
pdscli {request | submit} GetSlammerStatistics
   ID=GUID-or-FQN-of-Slammer
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all Slammers for which you want to display performance statistics.

EXAMPLES Run the following command to display performance statistics for both Slammers that are configured on your system.

```
pdscli request GetSlammerStatistics @input
```

where input.req contains the following name/value pairs:

```
Request=GetSlammerStatistics
ID=/Company-XZY-data/Slammer1
ID=/Company-XZY-data/Slammer2
```

See the GetSlammerStatisticsResponse.txt file for example response data.

SEE ALSO GetAllSlammers on p. 231

GetSlammerDetails on p. 295

Ranges for Option Values on p. 57

NAME **GetSnapLUNDetails**

DESCRIPTION Displays information for a specified Snap LUN.

SYNTAX

```
pdscli {request | submit} GetSnapLUNDetails
   Source=GUID-or-FQN-of-schedule
```

OPTIONS

Source

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the LUN for which you want to display information.

EXAMPLES Run the following command to display information for the SLUN_01 Snap LUN

```
pdscli request GetSnapLUNDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetSnapLUNDetails
Source=/SLUN_01
```

See the `GetSnapLUNDetails`Response.txt file for example response data.

SEE ALSO

DeleteSnapLUN on p. 191

GetAllSnapLUNs on p. 233

GetAllSnapLUNHierarchies on p. 232

ModifySnapLUN on p. 381

PerformLUNDuplication on p. 447

PerformBackgroundLUNDuplication on p. 402

Ranges for Option Values on p. 57

NAME **GetSNMPConfigDetails**

DESCRIPTION Displays information about the Simple Network Management Protocol (SNMP) configuration settings.

SYNTAX

```
pdscli {request | submit} GetSNMPConfigDetails
```

OPTIONS None.

EXAMPLES Run the following command to display information about the SNMP configuration on your system.

```
pdscli request GetSNMPConfigDetails
```

See the GetSNMPConfigDetailsResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **GetSpareDriveDetails**

DESCRIPTION Displays information about the spare disk drives in the Bricks that are configured on the system.

SYNTAX

```
pdscli {request | submit} GetSpareDriveDetails
```

OPTIONS None.

EXAMPLES Run the following command to display information about the spare disk drives in the Bricks that are configured on your system.

```
pdscli request GetSpareDriveDetails
```

See the GetSpareDriveDetailsResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **GetStorageConfig**

DESCRIPTION Displays information about the storage configuration on the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetStorageConfig
```

OPTIONS None.

EXAMPLES Run the following command to display information about your storage configuration.

```
pdscli request GetStorageConfig
```

See the GetStorageConfigResponse.txt file for example response data.

SEE ALSO GetNASConfigDetails on p. 274

GetSystemStatusDetails on p. 303

Ranges for Option Values (p. 57)

NAME **GetStorageRedundancyDetails**

DESCRIPTION Displays results from the most recent run of the PerformVerifyStorageRedundancy request.

SYNTAX

```
pdscli {request | submit}
   GetStorageRedundancyDetails
```

OPTIONS None.

EXAMPLES Run the following command to display results of the latest storage redundancy verification.

```
pdscli request GetStorageRedundancyDetails
```

See the GetStorageRedundancyDetailsResponse.txt file for example response data.

SEE ALSO PerformVerifyStorageRedundancy on p. 516

Ranges for Option Values on p. 57

NAME **GetSystemInformationFileDetails**

DESCRIPTION Displays results from the most recent run of the PerformVerifySystemConfiguration request.

SYNTAX

```
pdscli {request | submit}
  GetSystemInformationFileDetails
```

OPTIONS None.

EXAMPLES Run the following command to display results of the latest system configuration verification.

```
pdscli request GetSystemInformationFileDetails
```

See the GetSystemInformationFileDetailsResponse.txt file for example response data.

SEE ALSO PerformCollectSystemInformation on p. 407

GetFileSystemStatistics on p. 260

Ranges for Option Values on p. 57

NAME

GetSystemStatusDetails

DESCRIPTION

Displays summary status information for the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetSystemStatusDetails
  RequestForEC={true | false}
```

OPTIONS

RequestForEC

EXAMPLES

Run the following command to display status information for your system.

```
pdscli request GetSystemStatusDetails
  RequestForEC=True
```

See the GetSystemStatusResponse.txt file for example response data.

SEE ALSO

GetAllTasks on p. 235

GetAllSessions on p. 230

PerformCollectSystemInformation on p. 407

Ranges for Option Values on p. 57

NAME **GetTaskDetails**

DESCRIPTION Displays information about the specified tasks. A *task* is a fundamental unit of work within a Pillar Axiom storage system. The system converts every configuration request into one or more tasks, queues the tasks so that dependencies are satisfied, and performs the tasks.

SYNTAX

```
pdscli {request | submit} GetTaskDetails
  Task=GUID-or-FQN-of-task
```

OPTIONS **Task**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all tasks for which you want to display information.

EXAMPLES Run the following command to display information about two configuration tasks.

```
pdscli request GetTaskDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetTaskDetails
Task[6]=IDA0479DC2-38EF-9828-8902-
  E0FC03743270
Task[7]=IDf1234567-89ab-cde0-0123-
  456789eeeee0
```

See the GetTaskDetailsResponse.txt file for example response data.

SEE ALSO GetAllTasks on p. 235

Ranges for Option Values on p. 57

NAME **GetUpdatePackageDetails**

DESCRIPTION Displays information about the specified update packages that are staged on the Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} GetUpdatePackageDetails
  UpdatePackage=GUID-or-FQN-of-upgrade-
    package
```

OPTIONS **UpdatePackage**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of all update packages for which you want to display information.

For simplicity, use a meaningful name such as /Current, /Previous, or / Staged. For example, if you have just uploaded an update package, you can display the details and components of that recently-staged package with:

GetUpdatePackageDetails UpdatePackage=/Staged.

EXAMPLES Run the following command to display information about an update package that has been staged, but not yet installed, on your system.

```
pdscli request GetUpdatePackageDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetUpdatePackageDetails
UpdatePackage=/Staged
```

See the GetUpdatePackageDetailsResponse.txt file for example response data.

SEE ALSO GetAllUpdateComponents on p. 236

GetPilotDetails on p. 280

GetSlammerDetails on p. 295

GetBrickDetails on p. 242

Ranges for Option Values (p. 57)

NAME **GetVirtualInterfaceDetails**

DESCRIPTION Displays information about the specified secondary virtual interfaces that are defined for a File Server. The primary virtual interface is an attribute of the File Server itself.

SYNTAX

```
pdscli {request | submit}
  GetVirtualInterfaceDetails
    VirtualInterface=GUID-or-FQN-of-interface
```

OPTIONS **VirtualInterface**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all virtual interfaces for which you want to display information.

EXAMPLES Run the following command to display information about a secondary virtual interface that you defined for the Company-XYZ-FS1 File Server.

```
pdscli request GetVirtualInterfaceDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetVirtualInterfaceDetails
VirtualInterface=IDA0479DC2-38EF-9828-8902-
  E0FC03743270
```

See the GetVirtualInterfaceDetailsResponse.txt file for example response data.

SEE ALSO GetAllVirtualInterfaces on p. 239

Ranges for Option Values on p. 57

NAME **GetVolumeGroupDetails**

DESCRIPTION Displays information about the specified volume groups.

SYNTAX

```
pdscli {request | submit} GetVolumeGroupDetails
   VolumeGroup=GUID-or-FQN-of-volume-group
```

OPTIONS **VolumeGroup**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of all volume groups for which you want to display information.

EXAMPLES Run the following command to display information about the Eng volume group.

```
pdscli request GetVolumeGroupDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetVolumeGroupDetails
VolumeGroup=/Eng
```

See the GetVolumeGroupDetailsResponse.txt file for example response data.

SEE ALSO GetAllVolumeGroups on p. 240

Ranges for Option Values on p. 57

NAME

Login

DESCRIPTION

Starts a command line interface (CLI) session for the specified Pillar Axiom administrator or NDMP account.

SYNTAX

```
pdscli {request | submit} Login
   Username=string
   Password=string
```

OPTIONS

Username

Enter a valid user name that is defined for a Pillar Axiom administrator account.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

Password

Enter a valid password that is defined for the specified user name.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

EXAMPLES

Run the following command to log in to manage your Pillar Axiom storage system.

```
pdscli request Login @input
```

where input.req contains the following name/value pairs:

```
Request=Login
Username=xavier
Password=s7Z12W79
```

See the LoginResponse.txt file for example response data.

SEE ALSO

Logout on p. 310

Relogin on p. 524

Ranges for Option Values on p. 57

NAME **Logout**

DESCRIPTION Terminates a command line interface (CLI) session.

SYNTAX

```
pdscli {request | submit} Logout
```

OPTIONS None.

EXAMPLES Run the following command to log out from your administrative session.

```
pdscli request Logout
```

See the LogoutResponse.txt file for example response data.

SEE ALSO Login on p. 309

Relogin on p. 524

Ranges for Option Values on p. 57

NAME **ModifyAccount**

DESCRIPTION Modifies the specified administrator account.

SYNTAX

```
pdscli {request | submit} ModifyAccount
   Account=GUID-or-FQN-of-account
   Username=login-name
   Fullname=First Last
   Password=new-password
   RetypePassword=new-password
   EmailAddress=name@domain.com
   PhoneNumber=telephone-number
   Status={Enabled | Disabled}
   Role={Type1 | Type2 | Type3}
   SSHState={Enabled | Disabled}
```

OPTIONS

Account

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the account to modify.

Username [optional]

Identifies the administrator's login (user) name.

Fullname [optional]

Identifies the administrator's full name ("first" and "last").

Password [optional]

Identifies the login password. Passwords are case sensitive, and blank passwords are not permitted.

RetypePassword [optional]

Enter the password again.

EmailAddress [optional]

Identifies the recipient's email address. The email server to which the Pillar Axiom system sends alerts must be able to send messages to this email address.

PhoneNumber [optional]

Identifies the recipient’s telephone number. The Pillar Axiom system does not verify the validity of this entry.

Note: Enclose telephone numbers with spaces in quotes.

Status [optional]

Identifies whether the administrator account is enabled.

- Enabled accounts are active. Administrators whose accounts are enabled can log in to the Pillar Axiom storage system.
- Disabled accounts are inactive. Administrators whose accounts are disabled cannot log in.

Role [optional]

Identifies the administrator’s role. A role defines the administrator’s privileges. Choose from:

- Type1, if the person can perform all configuration and administration tasks.
- Type2, if the person can perform all tasks except create, modify, and delete administrator accounts and File Servers; modify global, SNMP, and NDMP settings; modify software or hardware configurations; or shut down the system.
- Type3, if the person can display information only, and cannot modify the configuration.

You cannot assign administrators to the following predefined roles:

- PrimarySystem, which identifies the Primary system administrator. This role has the same privileges as the Type1 role.
- PrimarySupport, which identifies the Primary Support Administrator. This role can perform support-related activities.

You cannot delete the PrimarySystem and PrimarySupport accounts.

SSHState [optional]

Controls whether or not the account will allow the Support account to enable SSH with a support request.

- Enabled allows the Support account to enable SSH. The Admin1 cannot enable SSH under any circumstances.
- Disabled prohibits the Support account to SSH.

EXAMPLES

Run the following command to disable Yvette Smith’s account when she starts a three-month leave of absence.

```
pdscli request ModifyAccount @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyAccount
Account=IDf1234567-89ab-cde0-0123-456789eeeee0
```

```
Username=Yvette
Status=Disabled
SSHState=Enabled
```

See the ModifyAccountResponse.txt file for example response data.

SEE ALSO

CreateAccount on p. 92

DeleteAccount on p. 169

ModifyMyAccount on p. 367

Ranges for Option Values on p. 57

NAME **ModifyBrick**

DESCRIPTION Modifies the specified Brick component.

SYNTAX

```
pdscli {request | submit} ModifyBrick
  Brick=GUID-or-FQN-of-Brick
  Name=string
```

OPTIONS

Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Brick to modify.

Name

Identifies the name that is assigned to a hardware component. Assign unique, meaningful component names to help you more easily locate specific components. The Pillar Axiom system maps the assigned name to the component's serial number and updates the map if you modify the component name.

EXAMPLES Run the following command to assign a new name to a Brick in your system.

```
pdscli request ModifyBrick @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyBrick
Brick=SE1
Name=Brick1
```

See the ModifyBrickResponse.txt file for example response data.

SEE ALSO GetAllBricks on p. 204

GetBrickDetails on p. 242

Ranges for Option Values on p. 57

NAME

ModifyCIFSShare

DESCRIPTION

Modifies the specified Common Internet File System (CIFS) share.

SYNTAX

```
pdscli {request | submit} ModifyCIFSShare
   CIFSShare=GUID-or-FQN-of-CIFS-share
   Name=share-name
   Comment=share-comment
   Enabled={true | false}
   Path=/path/to/directory/with/UNIX/slashes
```

OPTIONS

CIFSShare

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the CIFS share to modify.

Name [optional]

Identifies the name of a CIFS share. A filesystem must be shared before users can create or access data files. Share names must be unique within a File Server.

Comment [optional]

Describes the CIFS share.

Enabled [optional]

Identifies whether the CIFS share is enabled.

- Enabled shares are active. Users can access an enabled share point.
- Disabled shares are inactive. Users cannot access a disabled share point.

Path [optional]

Enter the full path to the CIFS share.

EXAMPLES

Run the following command to temporarily disable a specific CIFS share on the EngAtl filesystem.

```
pdscli request ModifyCIFSShare @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyCIFSShare
CIFSShare=IDA0479DC2-38EF-9828-8902-E0FC03743270
Enabled=false
```

See the ModifyCIFSShareResponse.txt file for example response data.

SEE ALSO

CreateCIFSShare on p. 95

DeleteCIFSShare on p. 171

Ranges for Option Values on p. 57

NAME

ModifyCloneFS

DESCRIPTION

Modifies a Clone FS on a Pillar Axiom system.

A Clone FS is a point-in-time, writeable snapshot of a filesystem, which can be used later to revert the filesystem. This snapshot retains the same QoS parameters as the source filesystem. It consumes storage capacity from the Clone FS storage space created for the source filesystem. A Clone FS can be scheduled to occur at any time. This provides a convenient method to branch from the source data without the need to do a full block-level copy.

An immediate Clone FS consumes part of the filesystem's capacity.

SYNTAX

```
pdscli {request | submit} ModifyCloneFS
  CloneFSFQN=FQN-of-clonefs
  CloneFSName=clonefs-name
  RequestedCapacity=integer
```

OPTIONS

CloneFSFQN

Identifies a specific Pillar Axiom object.

- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name.

  FQN Format:
  `/Fully/Qualified/ObjectName`

  Enter the FQN of the filesystem with which this Clone FS is associated. If there is more than one file server, use the FQN that includes the file server and filesystem.

Enter the GUID or FQN of the Clone FS you wish to modify.

CloneFSName [optional]

Identifies the name that is assigned to a Clone FS for administrative purposes. Clone FS names must be unique across the Pillar Axiom system.

RequestedCapacity

Identifies the requested capacity that is assigned to the object. This value is a soft limit, which means that data can be stored in a logical volume until the maximum capacity is reached.

EXAMPLES

Run the following command to modify the Clone FS for Company-XYZ.

```
pdscli request ModifyCloneFS @input
```

```
where input.req contains the following name/value pairs:
Request=ModifyCloneFS
CloneFSFQN=/Company-XYZ-FS1
CloneFSName=EngAtl
```

```
RequestedCapacity=500
```

See the ModifyCloneFSResponse.txt file for example response data.

SEE ALSO

CreateCloneFS on p. 97

DeleteCloneFS on p. 172

GetAllSnapLUNs on p. 233

GetAllCloneFSs on p. 208

GetAllCloneFSHierarchies on p. 207

NAME **ModifyExternalUPSDevice**

DESCRIPTION Modifies the specified external UPS device.

SYNTAX

```
pdscli {request | submit} ModifyExternalUPSDevice
   ups=GUID-or-FQN-of-Slammer
   IPAddress=string
   SNMPCommunityString=
```

OPTIONS

ups

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the UPS device to modify.

IPAddress [optional]

Identifies the IP address that is assigned to the External UPS Device.

SNMPCommunityString [optional]

Identifies a community for which a specific trap host should receive traps that the Pillar Axiom system generates. You can specify different community strings for each trap host so that multiple administrators can receive specific types of SNMP traps.

Default: public (note the lower-case spelling)

EXAMPLES Run the following command to modify the external UPS device.

```
pdscli request ModifyExternalUPSDevice @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyExternalUPSDevice
ups=UPS1
IPAddress=172.18.65.128
SNMPCommunityString=public
```

See the ModifyExternalUPSDeviceResponse.txt file for example response data.

SEE ALSO GetExternalUPSDeviceDetails on p. 250

DeleteExternalUPSDevice on p. 174

Ranges for Option Values on p. 57

NAME **ModifyFileServer**

DESCRIPTION Modifies the specified File Server.

SYNTAX

```
pdscli {request | submit} ModifyFileServer
  FileServer=GUID-or-FQN-of-File-Server
  Name=File-Server-name
  Comment=File-Server-comment
  PrimaryVirtualInterface.IPAddress=
    IP-address
  PrimaryVirtualInterface.IPNetmask=
    IP-netmask
  PrimaryVirtualInterface.VLANTag={0 |
    integer-1-through-4095}
  PrimaryVirtualInterface.FrameSize=integer
  PrimaryVirtualInterface.
    NetworkInterfaceIOPort.
    Slammer=GUID-or-FQN-of-Slammer
  PrimaryVirtualInterface.
    NetworkInterfaceIOPort.
    ControlUnitNumber={0 | 1}
  PrimaryVirtualInterface.
    NetworkInterfaceIOPort.
    EthernetPortName={PORT0 | PORT1}
  DefaultRoute.GatewayIPAddress=IP-address
  DNSConfig.DomainName=string
  DNSConfig.Server[1-3]IPAddress=IP-address
  NISConfig.DomainName=string
  NISConfig.ServerIPAddress=
    IP-address
  NSSSearchOrder.HostLookupService[1-2]=
    {NIS | DNS | None}
  NSSSearchOrder.PasswordLookupService[1-2]=
    {NIS | Files | None}
  AccountMapping.MapIdenticalNames=
    {true | false}
  AccountMapping.UseAllDomains=
    {true | false}
  AccountMapping.DomainName=string
  CIFSServer.ServerName=CIFS-server-name
  CIFSServer.Comment=CIFS-server-comment
  CIFSServer.CharacterSet={STANDARD-ASCII |
    IBM-437 | IBM-850}
  CIFSServer.OpportunisticLockingEnabled=
    {true | false}
  CIFSServer.WINSServer[1-3]IPAddress=
       IP-address
  CIFSServer.DomainName=string
  CIFSServer.UseAnonymousPDCAccess=
    {true | false}
  CIFSServer.PDCAccessAccount.DomainName=
    string
  CIFSServer.PDCAccessAccount.Username=
    string
  CIFSServer.PDCAccessAccount.Password=
    string
  NFSServer.PortNumber=integer
  NFSServer.CharacterSet={Standard-ASCII |
    UTF8 | ISO8859-1}
  NFSServer.AnonymousUID=integer-negative-OK
  NFSServer.AnonymousGID=integer-negative-OK
  NFSServer.MountReservePortOnly=
    {true | false}
  NFSServer.AllowNonRootChown={true | false}
  NFSServer.TCPSupported={true | false}
  NFSServer.TCPConnectionsLimit=integer
  CIFSRemove={true | false}
  NFSRemove={true | false}
  NFSCharacterSet={Standard-ASCII |
    UTF8 | ISO8859-1}
```

OPTIONS

FileServer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the File Server to modify.

Name [optional]

Identifies the name that is assigned to a File Server. File Server names must be unique on the Pillar Axiom storage system.

Comment [optional]

Describes the File Server.

PrimaryVirtualInterface.IPAddress [optional]

Identifies the IP address that is assigned to the primary virtual network interface.

PrimaryVirtualInterface.IPNetmask [optional]

Identifies the subnet mask that is assigned to the File Server's virtual network interface.

PrimaryVirtualInterface.VLANTag [optional]

Identifies the virtual LAN (VLAN) ID that is assigned to the File Server's virtual network interface.

- Assign a value of 1 through 4094, inclusive, if you have connected a VLAN-capable switch to the Pillar Axiom system.
- Leave the field blank to disable VLAN tagging.

PrimaryVirtualInterface.FrameSize [optional]

Identifies the packet frame size. This value defines the maximum transmission unit (MTU).

The frame size (MTU) does not include the Ethernet header portion of the packet. If your network switch has trouble with this, you can set the switch to a larger value or lower the MTU size to correct the problem.

- If your network supports extended Ethernet (jumbo) frames, enter an integer greater than 1500 and less than 16362. Make sure that this Pillar Axiom MTU size matches the network MTU size. If the MTU sizes are mismatched, users may experience I/O "hangs" when the client machines try to process too-large packets.
- If your network does not support jumbo frames, enter the default frame size of 1500.

PrimaryVirtualInterface.NetworkInterfaceIOPort. Slammer [optional]

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer that is the File Server's primary virtual interface.

PrimaryVirtualInterface.NetworkInterfaceIOPort. ControlUnitNumber [optional]

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs. The CUs are identified by 0 (the top CU) and 1 (the bottom CU).

PrimaryVirtualInterface.NetworkInterfaceIOPort. EthernetPortName [optional]

Identifies the port on the Slammer that is assigned to the File Server's virtual network interface. Select a gigabit Ethernet network port on on one of the data-path Slammers control unit (CU) .

- If Link Aggregation is enabled, select port 0 (zero). You cannot create virtual interfaces on alternate port 1 when Link Aggregation is enabled.
- If Link Aggregation is disabled, select either port.

DefaultRoute.GatewayIPAddress [optional]

Identifies the IP address that is assigned to the gateway host. The gateway IP address is used to route messages from this network to other networks.

DNSConfig.DomainName [optional]

Identifies a Domain Name Service (DNS) domain, which specifies the domain to be searched.

DNSConfig.Server[1-3]IPAddress [optional]

Identifies the IP address of the Domain Name Service (DNS) server in your network that translates host names to IP addresses.

NISConfig.DomainName [optional]

Identifies a Network Information Service (NIS) domain, which is a group of computers that share the same configuration information.

NISConfig.ServerIPAddress [optional]

Identifies the IP address of a Network Information Service (NIS) server that holds a database of user accounts (among other NIS-related entries). When a user attempts to access the Pillar Axiom system, the user is authenticated against the database.

NSSSearchOrder.HostLookupService[1-2] AND NSSSearchOrder.PasswordLookupService[1-2] [optional]

Identifies the search order that the Pillar Axiom system uses to resolve hosts and passwords. Choose from:

- NIS (if you use Network Information Services), DNS (if you use Domain Name System), or None to resolve hosts.
- NIS (if you use Network Information Services), Files (if you use /etc/passwd, /etc/group, and
  /etc/netgroup files), or None to resolve passwords.

If you want the system to search only one source for host resolution, password resolution, or both, enter None for the second search target.

AccountMapping.MapIdenticalNames [optional]

Identifies whether support for mapping CIFS-to-NFS user accounts is enabled.

- Enable CIFS-to-NFS mapping if you have a single authority to assign users with both an NFS account and CIFS account using the same name. Users can use NFS only, CIFS only, or both NFS and CIFS.
- Disable CIFS-to-NFS mapping if you don't have a single authority to assign NFS and CIFS accounts.

You should disable account mapping when:

- Users do not have both NFS and CIFS accounts.
- NFS and CIFS users account names are different.
- The authority to assign account names can be done by different groups.

Note that the user’s NFS account is the owner of files and directories that the user creates.

AccountMapping.UseAllDomains [optional]

Identifies whether CIFS-to-NFS account mapping is enabled for all CIFS domains.

- Enable this option to use CIFS-to-NFS account mapping for all CIFS domains.
- Enable this option to use CIFS-to-NFS account mapping for a single CIFS domain that you specify.

AccountMapping.DomainName [optional]

Identifies the CIFS domain name. Enter the NetBIOS name of the CIFS domain that contains definitions for authentication and account mapping.

Enter a CIFS domain name when you enable MapIdenticalNames and disable UseAllDomains.

CIFSServer.ServerName [optional]

Identifies the name by which users refer to the CIFS server. The CIFS server advertises its services by this name through the NetBIOS Naming Service, so this value is sometimes called the NetBIOS name.

CIFSServer.Comment [optional]

Describes the CIFS server. Clients who browse or list shares may view these comments.

CIFSServer.CharacterSet [optional]

Identifies which character set is valid. Choose from:

- Standard-ASCII, which is a 7-bit character set that includes 128 printable and control character values (0-127).
- IBM-437, which is an 8-bit character set that includes the Standard-ASCII values plus values from OEM code page 437. Used on most PCs that are sold in the USA.

- IBM-850, which is an 8-bit character set that includes the Standard-ASCII values plus values from OEM code page 850. Used on most PCs that are sold in Western European countries.

CIFSServer.OpportunisticLockingEnabled [optional]

Identifies whether opportunistic locks (oplocks) are enabled to cache client data.

- Read caching permits data that is stored on the server to be copied to the client. When a client application requests the data, the read request is satisfied by reading the client's copy of the data. Read caching can last forever. If the data that is stored on the server changes, the client is informed and the read cache is invalidated.
- Write caching permits data changes to be buffered in the client for up to two minutes. When a client application changes the data, the write request is satisfied by changing the client's copy of the data. Client-server connection problems that occur during the two-minute window can result in lost changes. If the client is still running, a popup message alerts the client to the lost changes.

Disable oplocks to require that all client requests must access a data file instead of cache.

CIFSServer.WINSServer[1-3]IPAddress [optional]

Identifies a Windows Internet Naming Service (WINS) server in your network. The WINS server is the central repository for Windows server names. Broadcasts of NetBIOS names do not cross subnets, so the WINS server can resolve names across multiple subnets.

Note: If you enable Active Directory for authentication, you do not need to enter a WINS server address.

CIFSServer.UseAnonymousPDCAccess [optional]

Identifies whether the File Server communicates with the CIFS Domain Controller as an anonymous user to authenticate users and to perform user-name-to-SID mappings (SID = security identifier).

- Enable this option if your Domain Controller is installed on a pre-Windows 2000 server or a Windows 2000 server that is a member of a pre-Windows 2000 domain. These configurations permit access by anonymous users.
- Disable this option if your Domain Controller is installed on a Windows Server 2003 machine. That configuration denies access by anonymous users.

CIFSServer.PDCAccessAccount.DomainName [optional]

Identifies the name of a Windows Domain Controller that maintains a database of user, group, and machine accounts. Enter the Domain Controller's name. Ensure that the Domain Controller is registered with the WINS or DNS servers that you specify. The Pillar Axiom system uses WINS or DNS, in that order, to resolve the name and locate the Domain Controller.

In this release, you must log in to a Windows 2000 native-mode domain controller as an administrator, run the following command, and reboot the domain controller:

```
net localgroup "Pre-Windows 2000 Compatible Access" everyone /add
```

This command relaxes a security restriction in Windows 2000 servers to disallow anonymous access to user lists and groups.

If you use a Windows 2000 mixed-mode domain controller for CIFS authentication, you do not have to run the command and reboot.

CIFSServer.PDCAccessAccount.Username [optional]

Identifies the administrator user name for the Domain Controller that is associated with the Pillar Axiom storage system.

CIFSServer.PDCAccessAccount.Password [optional]

Identifies the administrator password for the CIFS Domain Controller.

NFSServer.PortNumber [optional]

Identifies which port is reserved for NFS requests. If you allow NFS requests only from a secure port, enter a port number that is less than or equal to 1024.

Default: port 2049

NFSServer.CharacterSet [optional]

Identifies which character set is valid. Choose from:

- Standard-ASCII, which is a 7-bit character set that includes 128 printable and control character values (0-127).
- UTF8, which is a Unicode-coded character set that uses an 8-bit transformation format. Defined in RFC 2279.
- ISO8859-1, which is an 8-bit, single-byte character set that is an extension of Standard-ASCII.

NFSServer.AnonymousUID [optional]

Identifies the user ID (UID) for anonymous users.

- Set the UID to zero (0) if you want anonymous-user access to be identified as the root user. This is a very insecure setting. Setting anonymous to root allows all users full control.
- Set the UID to the “nobody” value (often -2, but not always) if you want anonymous-user access to be identified as the user “nobody.” In most instances, this is the recommended setting.
- Set the UID to any other user ID that you want to be identified as anonymous-user access.

NFSServer.AnonymousGID [optional]

Identifies the group ID (GID) of a file where the group owner is not an NFS group for example, a CIFS group.

NFSServer.MountReservePortOnly [optional]

Identifies whether all NFS mount requests must be received on a secure port that is reserved for NFS requests only.

- Enable the option if NFS mount requests must be received on a secure port that is reserved for NFS requests (including mount requests). The port number must be less than 1024.
- Disable the option if NFS mount requests can be received on any open port.

NFSServer.AllowNonRootChown [optional]

Identifies whether users who do not have root privileges can change the owner of files that are stored in the Pillar Axiom system.

- Enable the option if non-root users can change ownership on only files that they own.
- Disable the option if only root users can change file ownership.

Note: When quotas are enabled, you may set this flag to automatically transfer files to other users if an individual user exceeds a quota limit.

NFSServer.TCPSupported [optional]

Identifies whether Transmission Control Protocol (TCP) connections are allowed.

- Enable TCP if users can mount filesystems using the Transmission Control Protocol (TCP).
- Disable TCP if users can mount filesystems using only the User Datagram Protocol (UDP).

NFSServer.TCPConnectionsLimit [optional]

Identifies how many concurrent protocol-specific connections are allowed. Enter a value that is large enough to meet your anticipated work load.

- A lower number allows fewer users to access data simultaneously. Once the limit is reached, access to additional users is denied. Set this value to a lower number only if there is a specific reason to restrict the number of NFS TCP connections.
- A higher number allows more users to access data simultaneously. There is no performance penalty associated with setting this value to the maximum (65,000).

CIFSRemove [optional]

Identifies whether to remove all CIFS configurations and associated shares that currently exist on this File Server.

- Enable this option to remove the CIFS configurations and all CIFS configuration details.
- Disable this option to keep the CIFS configurations and the CIFS configuration details.

NFSRemove [optional]

Identifies whether to remove all NFS configurations and associated exports that currently exist on this File Server. After the NFS configurations are removed, NFS users no longer have access to the File Server.

- Enable this option to remove the NFS configurations and all NFS configuration details.
- Disable this option to keep the NFS configurations and the NFS configuration details.

NFSCharacterSet [optional]

Identifies which character set is valid. Choose from:

- Standard-ASCII, which is a 7-bit character set that includes 128 printable and control character values (0-127).
- UTF8, which is a Unicode-coded character set that uses an 8-bit transformation format. Defined in RFC 2279.
- ISO8859-1, which is an 8-bit, single-byte character set that is an extension of Standard-ASCII.

EXAMPLES

Run the following command to increase the TCP connection limit for NFS users.

```
pdscli request ModifyFileServer @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyFileServer
FileServer=/Company-XYZ-FS1
NFSServer.TCPConnectionsLimit=1250
```

See the ModifyFileServerResponse.txt file for example response data.

SEE ALSO

CreateFileServer on p. 100

DeleteFileServer on p. 175

Ranges for Option Values on p. 57

NAME **ModifyFileSystem**

DESCRIPTION Modifies the specified filesystem.

SYNTAX

```
pdscli {request | submit} ModifyFileSystem
   FileSystem=GUID-or-FQN-of-filesystem
   VolumeGroup=GUID-or-FQN-of-volume-group
   Capacity.CurrentCapacity=integer-0-through-2147483647
   Capacity.MaximumCapacity=integer-1-through-2147483647
   RetentionPolicy.MinimumRetentionPeriod.Units=
      {Years | Months | Days}
   RetentionPolicy.MinimumRetentionPeriod.Amount=integer
   RetentionPolicy.MaximumRetentionPeriod.Units=
      {Years | Months | Days}
   RetentionPolicy.MaximumRetentionPeriod.Amount=integer
   RetentionPolicy.DefaultRetentionPeriod.Units=
      {Years | Months | Days}
   RetentionPolicy.DefaultRetentionPeriod.Amount=integer
   RetentionPolicy.AllowExpiredFileDeletion={true | false}
   RetentionPolicy.EnforceVolumeRetention={true | false}
   Profile={OracleASM | HighThroughput | Normal}
   RelativePriority={Premium | High | Medium | Low | Archive}
   FileSizeBias={None | Large | Medium | Small}
   AccessBias={Sequential | Random | Mixed | None}
   IOBias={Read | Write | Mixed | None}
   TemporaryPriority={Premium | High | Medium | Low | Archive}
   StorageClass={MultiLevelCellSolidStateDrives |
   SingleLevelCellSolidStateDrives | FibreChannelHardDiskDrives |
   SataHardDiskDrives}
   Redundancy={Standard | Double}
   CloneFSCapacity=integer
   AuditingMode={true | false}
   AutomaticallyAssignSlammer={true | false}
   Slammer=GUID-or-FQN-of-Slammer
   ControlUnitNumber={0-for-automatic-assignment | integer}
   Name=New-Filesystem-Name
   BackgroundCopyPriority=
   {MaximizeSpeed | MinimizeImpact | SystemChooses}
```

OPTIONS

FileSystem

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem to modify.

VolumeGroup [optional]

Enter the GUID or FQN of the volume group with which the filesystem is associated.

Capacity.CurrentCapacity [optional]

Identifies the initial capacity, in MB, that is assigned to the object. This value is a soft limit, which means that data can be stored in a logical volume until the maximum capacity is reached.

Capacity.MaximumCapacity [optional]

Identifies the maximum capacity, in MB, to which a given object can grow.

RetentionPolicy.MinimumRetentionPeriod.Units

The Retention Policy settings specify the period for which files on the SecureWORMfs must be retained. The MinimumRetentionPeriod (Units) specifies the minimum retention period. Valid values are Years, Months, or Days.

RetentionPolicy.MinimumRetentionPeriod.Amount

Specifies the minimum retention period. Valid values are 0 to 2147483647.

To define an unlimited value, enter 0 (zero).

RetentionPolicy.MaximumRetentionPeriod.Units

Specifies the maximum retention period. Valid values are Years, Months, or Days.

RetentionPolicy.MaximumRetentionPeriod.Amount

Specifies the minimum retention period. Valid values are 0 to 2147483647 and must be greater than or equal to the minimum retention value.

To define an unlimited value, enter 0 (zero).

RetentionPolicy.DefaultRetentionPeriod.Units

Specifies the maximum retention period for the SecureWORMfs. Valid values are Years, Months, or Days.

RetentionPolicy.DefaultRetentionPeriod.Amount

Specifies the minimum retention period. Valid values are 0 to 2147483647 and must be greater than or equal to the minimum and less than or equal to the maximum retention value.

To define an unlimited value, enter 0 (zero).

RetentionPolicy.AllowExpiredFileDeletion

Allows you to delete the SecureWORMfs if it has reached the specified retention policy setting expiration and is therefore no longer protected.

RetentionPolicy.EnforceVolumeRetention

Forces the retention policy settings for a SecureWORMfs.

Files written over CIFS connections are protected immediately, not allowing any further modifications.

Files written over NFS connections must be protected explicitly by setting the Immutable field for the file in its corresponding attribute file.

For instructions on how to edit the attribute file, refer to the *Pillar Axiom Administrator's Guide*.

Profile [optional]

Identifies a performance profile. Choose from:

- Normal, which defines specific performance and redundancy settings for each logical volume. This value is highly recommended and should be used in almost all cases.
- OracleASM, which uses a wide stripe RAID geometry consisting of 1 MB strips to improve the overall performance of logical volumes in an Oracle ASM environment.
- HighThroughput, which stripes the filesystem or LUN across all RAID arrays. SATA Bricks have two RAID groups per Brick, and FC Bricks have one RAID group per Brick. You do not need to set double redundancy to get full array striping.

Important! Use of the HighThroughput profile is for special purposes only. Do not use this profile for ordinary filesystems or LUNs. Please contact the Pillar Customer Services Advanced Solutions Group before using this profile option, which should only be used for special testing where the system configuration is cleared after the test.

RelativePriority

The Relative Priority determines how much of the system resources are devoted to that volume, including the allocation of Slammer CPU cycles and the allocation of specific portions of the disk platters. The higher the Priority, the greater the allocation of CPU time and the faster the media access time.

The Relative Priority option prioritizes the layout of data that is stored in the Pillar Axiom system based on one of the following settings:

- Premium priority is the highest possible performance. This priority uses the outer 20% of the drive platters.

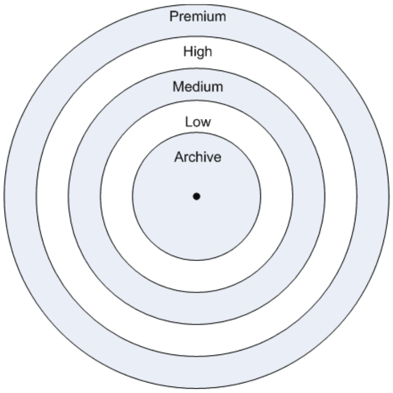
Note: I/O is typically faster when data resides on the outer edge of the drive platters.

- High priority logical volumes are allocated space in the outer 20%-40% of the drive platters.

Tip: If sufficient Bricks are present, performance of high priority volumes is enhanced by placing those volumes on a greater number of Bricks.

- Medium priority logical volumes are allocated space in the outer 40% to 60% of the drive platters.
- Low priority logical volumes are allocated space in the band that is 60% to 80% from the outer diameter of drive platters.
- Archive (lowest-priority) logical volumes are allocated space in the inner areas of the drive platters.

Figure 3 Example of HDD priority bands



If needed, the Pillar Axiom system allocates space in a higher or lower priority band (within the same Storage Class) than the one that you choose.

FileSizeBias

Identifies the typical size of files that are stored in the specified logical volume. Choose from:

- Small if files are smaller than 20 kilobytes (KB).
- Medium if files are larger than 20KB and smaller than 4 megabytes (MB).
- Large if files are larger than 4MB.

AccessBias

Identifies the typical data access method. Choose from:

Table 19 Data access methods

Access bias	I/O bias	Read ahead in the Slammer	RAID configuration in the Brick
Sequential	Read	Aggressive	RAID 5 Reads large extents of the data into memory.
	Write	Conservative	RAID 5 • Allocates capacity in large, full-stripe (640 K) extents. • Writes data to physical storage more quickly and in full-stripe extents.
Mixed and random	Read	None	RAID 5 • Reads small extents into memory. • Allocates capacity in small chunks. • Writes data to physical storage more slowly to combine (or eliminate) write operations.
Random	Write	None	RAID 10

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

IOBias

Identifies the typical read-write ratio. Choose from:

- Read if users or applications read data more often than they write to the data source.
- Write if users or applications write data more often than they read it.

Important! If you choose Random as the access method and Write as the I/O Bias, the system creates the volume with a Distributed RAID geometry. This geometry provides enhanced write performance but uses twice the capacity.

- Mixed if the read-write ratio varies.

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

The system stores all writes of user data and system metadata in mirrored copies of the journal. Although the filesystem or LUN is journaled, for some of the profiles, write caching is disabled because it tends to have very low hit rates.

One copy is maintained in non-volatile memory on one control unit (CU) of a Slammer. The mirror copy is maintained in one of:

- Battery-backed memory of the partner CU on the Slammer (preferred location). Writes to this copy are equivalent to write-back cache.
- An area reserved on disk for the logical volume if the partner CU is unavailable for the write. Writes to this copy are equivalent to write-through cache.

Writes from the journal to permanent disk storage are equivalent to write-through cache. The system flushes user data and the corresponding metadata as a unit to disk.

TemporaryPriority

Temporarily changes the Priority level of the filesystem without changing the physical location of the volume. When you want to change the Priority back to the original level, modify the Quality of Service (QoS) of the volume.

One use for temporarily changing the priority level is to allow for a restore operation or data load operation where the original QoS would have been set to optimize for Write Archive or Random Read or Write, respectively.

Important! If the temporary QoS change is not changed back to the original QoS, it can have adverse effects on other system resources. If the change is to be made permanent, the QoS should be changed and the data migrated.

StorageClass

Identifies the category of physical storage on which the logical volume resides:

- Fibre Channel drives
- SATA drives
- SATA single-level cell solid state drives
- SATA multi-level cell solid state drives

Redundancy

Identifies how many mirror copies of the original data are stored online.

Important! Pillar highly recommends that you consult with a Pillar customer support professional for assistance with sizing your system and creating your volumes (filesystems and LUNs).

Redundancy options include:

- Standard, to store original data only. Data striping over multiple RAID groups maintains full redundancy, even without mirror copies.
- Double, to store original data and one mirror copy, with data striping over multiple RAID groups.

Note: Double Redundancy can only provide true redundancy if your system has enough Bricks to allocate the filesystem or LUN such that no two mirror copies share a RAID group.

SATA Bricks have two RAID groups per Brick, FC Bricks have one RAID group per brick.

If the storage pool is becoming depleted, or a large filesystem or LUN is created, it might be necessary to place the filesystem or LUN on more RAID group fragments.

Depending on the system’s ability to allocate sufficient contiguous storage blocks for the size of the logical volume, use the following number of RAID groups to configure your volumes for best performance:

Table 20 Optimum number of RAID groups for best performance

Priority	**SATA standard Redundancy**	**SATA double redundancy**	**FS standard redundancy**	**FS double redundancy**
Archive	4	8	2	4
Low	4	8	2	4
Medium	6	12	3	6
High	8	16	4	8
Premium	8	16	4	8

For *performance testing purposes only*, create a filesystem or LUN using standard redundancy and the High Throughput Profile. This is NOT recommended for most applications. Reset your system after you have created a High Throughput LUN or filesystem before you configure normal filesystems or LUNs for applications.

CloneFSCapacity

Specifies the maximum amount of repository space to be reserved for Clone FSs from the total system capacity.

⚠ CAUTION

We strongly recommend that you allocate sufficient repository capacity to minimize the chances of running out of this space (which could lead to data inconsistency or loss). To set sufficient capacity, use a value equal to the source volume capacity times the number of replicas times the maximum rate of change. For example, for a 100 GB volume that is projected to have 20 active replicas at a time and no more than a 20% rate of change, use a value of 400 GB for the clone repository.

Maximum capacity for Clone FSs is 2147483647.

AuditingMode

Enables the logging of compliance activities, including:

- The full path to a protected file and the mechanism used to protect the file (manual or automatic).
- Changes to the expiration time of a file. If expiration time is derived from the default retention period, the value of default retention period is also logged in addition to the new and old expiration times and the full path to the file.
- Deletion of an expired file. The log for this event contains the current system time, expiration time of the file, and the full path to the file.
- Detection of data inconsistency in a file. The actual and expected data hash values are logged.

All records in the audit log are in UTF-8 format and have a timestamp and date.

Slammer [optional]

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer with which the filesystem is associated.

Note: If you do not enter a value, the system will automatically designate a Slammer. If AutomaticallyAssignSlammer is false, the specified value changes the Slammer assignment.

ControlUnitNumber [optional]

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs. The CUs are identified by 0 (the top CU) and 1 (the bottom CU).

Note: If you do not enter 0 or 1, the system automatically designates a CU. If AutomaticallyAssignSlammer is false, the specified value changes the CU assignment.

Name

Identifies the new name that is assigned to a filesystem. Each filesystem name must be unique within its associated File Server and volume group.

BackgroundCopyPriority

Identifies the strategy the system should use to control the impact on performance when background tasks need to copy or move data from one location in the storage pool to another:

- MinimizeImpact. Restricts the amount of work performed on a loaded system. This option is intended to have a minimal impact on client I/O throughput at the expense of longer copy times.
- SystemChooses. Balances the background copy with the incoming client I/O. This option is the default.
- MaximizeSpeed. Prioritizes the background copy at the expense of client I/O throughput.

Note: When the system is idle or lightly loaded, the above background task maximizes the amount of work done regardless of the option selected.

The following types of operations are affected by the strategy you select:

- Copy
- Restore
- Quality of Service (QoS) Priority changes
- Storage Class changes

Data transfer operations invoked by the Pillar Axiom MaxRep Replication for NAS utility are not affected by the BackgroundCopyPriority setting.

EXAMPLES

Run the following command to increase the capacity that is assigned to the EngAtl filesystem.

```
pdscli request ModifyFileSystem @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyFileSystem
FileSystem=/Company-XYZ-FS1/EngAtl
VolumeGroup=/Eng
Capacity.CurrentCapacity=500
Capacity.MaximumCapacity=1000
RetentionPolicy.EnforceVolumeRetention=true
RetentionPolicy.MinimumRetentionPeriod.Amount=0
RetentionPolicy.MaximumRetentionPeriod.Profile=Normal
RelativePriority=High
FileSizeBias=Small
AccessBias=Sequential
IOBias=Write
StorageClass=FibreChannelHardDiskDrives
Redundancy=Double
```

See the ModifyFileSystemResponse.txt file for example response data.

SEE ALSO

CreateFileSystem on p. 110

DeleteFileSystem on p. 176

Ranges for Option Values on p. 57

NAME **ModifyFileSystemSnapshot**

DESCRIPTION Modifies the specified Snap FS of a filesystem. When you change the name of a Snap FS, its type is implicitly changed from system managed to administrator managed. If the Snap FS was created by a recurring schedule, the Snap FS is no longer considered in the schedule's rotation of older snapshots.

SYNTAX

```
pdscli {request | submit} ModifyFileSystemSnapshot
   ID=GUID-or-FQN-of-Snap-FS
   Name=string
```

OPTIONS

ID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Snap FS to modify.

Name

Identifies the name that is assigned to an administrator-managed Snap FS.

EXAMPLES Run the following command to rename a scheduled Snap FS that was created from the EngAtl filesystem.

```
pdscli request ModifyFileSystemSnapshot @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyFileSystemSnapshot
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
Name=EngAtl-snapshot-save
```

See the ModifyFileSystemSnapshotResponse.txt file for example response data.

SEE ALSO

ModifyJob on p. 345

CreateFileSystemSnapshotSchedule on p. 124

DeleteFileSystemSnapshot on p. 177

DeleteFileSystemSnapshotSchedule on p. 179

GetAllFileSystemSnapshots on p. 216

GetAllFileSystemSnapshotSchedules on p. 218

GetFileSystemSnapshotDetails on p. 256

GetFileSystemSnapshotScheduleDetails on p. 258

PerformFileSystemDuplication on p. 425

PerformBackgroundFileSystemDuplication on p. 400

PerformRestoreFileSystemFromSnapshot on p. 485

PerformRebuildFileSystemRedundancy on p. 474

Ranges for Option Values on p. 57

NAME

ModifyHBAPort

DESCRIPTION

Modifies the specified HBA port.

SYNTAX

```
pdscli {request | submit} ModifyHBAPort
   HBAPortWWN=World-wide-name
   HBAPortName=HBA-port-name
   SANHostName=Host-name
```

OPTIONS

HBAPortWWN

Identifies the address, or world wide name (WWN), of the specified host bus adapter (HBA).

HBAPortName [optional]

Identifies the name given to the gigabit interfce converter (GBIC), or port.

SANHostName [optional]

Identifies the name assigned to a specific SAN host.

EXAMPLES

Run the following command to modify an HBA port.

```
pdscli request ModifyHBAPort @input
```

where input.req contains the following name/value pairs:

```
HBAPortWWN=21:00:00:e0:8b:11:b5:4a
HBAPortName=HBA 3 Port 0
SANHostName=SANTest_02
```

See the ModifyHBAPortResponse.txt file for example response data.

SEE ALSO

GetSANConfigDetails on p. 284

GetHostMappingConfigDetails on p. 265

GetAllSANHosts on p. 227

GetSANHostDetails on p. 285

ModifySANHost on p. 377

GetHBAPortConnections on p. 262

DeleteSANHost on p. 189

NAME **ModifyiSCSIPortDetails**

DESCRIPTION Modifies the specified iSCSI port settings.

SYNTAX

```
pdscli {request | submit} ModifyiSCSIPortDetails
   Slammer=GUID-or-FQN-of-UPS-device
   ControlUnitNumber={0 | 1}
   iSCSIPortName={PORT0 | PORT1}
   iSCSITCPPort={0 | integer}
   MTUBytes={0 | integer}
   IPAssignment={Manual | DHCP}
   IPAddress=IP-address
   IPNetmask=IP-netmask
   GatewayIPAddress=IP-address
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

iSCSIPortName

Identifies the name (0 or 1) of the iSCSI port device.

iSCSITCPPort [Optional]

Idenfities the iSCSI port number.

MTUBytes [Optional]

Identifies the number of bytes of the MTU (the maximum transmission unit). The MTU is the largest physical packet size that a network can transmit. Any messages larger than the MTU (default is 1500) are divided into smaller packets before being sent.

IPAssignment [Optional]

DHCP (Dynamic Host Configuration Protocol) is a set of rules used to ensure that all IP addresses are unique. You can also assign IP addresses manually.

IPAddress [Optional]

Identifies the IP address that is assigned to the iSCSI port device.

IPNetmask [Optional]

Identifies a subnet mask for the range of IP addresses at the iSCSI port's destination.

GatewayIPAddress [Optional]

Identifies the IP address that is assigned to the gateway host. The gateway IP address is used to route messages from this network to other networks.

EXAMPLES

Run the following command to display Slammer command output.

```
pdscli request ModifyiSCSIPortDetails @input
```

where input.req contains the following name/value pairs:

```
Request=GetSlammerControlUnitDiagnostic
  Details
Slammer=/Company-XZY-data/Slammer1
ControlUnitNumber=1
iSCSIPortName=Port0
iSCSITCPPort=
MTUBytes=65600
IPAssignment=DHCP
IPAddress=172.20.67.0
IPNetmask=255.255.255.0
GatewayIPAddress=172.18.65.110
```

See the ModifyiSCSIPortDetails
Response.txt file for example response data.

See Also

GetAllSlammers on p. 231

GetFCPortStatistics Details on p. 251

GetFileServerDetails on p. 253

GetFileSystemDetails on p. 255

GetiSCSIPortStatisticsDetails on p. 266

GetSANConfigDetails on p. 284

GetSANSlammerPortDetails on p. 290

GetSlammerDetails on p. 295

NAME **ModifyJob**

DESCRIPTION Modifies the specified Pillar Axiom job.

SYNTAX

```
pdscli {request | submit} ModifyJob
  Job=GUID-or-FQN-of-job
  Name=string
  Schedule=GUID-or-FQN-of-schedule
  Enabled={true | false}
  CreateFileSystemSnapshot.Name=string
  CreateFileSystemSnapshot.FS=GUID-or-FQN-
    of-filesystem
  PerfromUpdate.UpdatePackage=
    .tar.gz-file-name
  PerformUpdate.UpdateComponentName[1-8]=
    {"Pilot Software" | "Pilot OS" |
    "Slammer Software" | "Slammer PROM" |
    "Slammer Software AX600" | "Slammer PROM AX600" |
    "Brick Firmware" | Brick FC Firmware}
  PerformUpdateDiskDrive.UpdatePackage=
    GUID-or-FQN-of-Brick
  PerformUpdateDiskDrive.UpdateComponentName
    =Name-of-updatable-component
  CreateQuota.AllowFileSystemOffline=
    {true | false}
  CreateQuota.FileSystem=GUID-or-FQN-of-
    filesystem
  CreateQuota.Path=/path/to/directory
  CreateQuota.RestrictUsers={AllAggregate |
    SingleUser | Group}
  CreateQuota.EnforceAllLimitsForDirectory=
    {true-if-AllAggregate | false-otherwise}
  CreateQuota.Username=string-if-select-
    SingleUser
  CreateQuota.Groupname=string-if-select-
    Group
  CreateQuota.SoftLimitInMB={0-for-unlimited
    |integer}
  CreateQuota.HardLimitInMB={0-for-unlimited
    |integer}
  CreateQuota.MBGracePeriodInDays=
    {0-for-unlimited | integer}
  DeleteQuota.ID=GUID-or-FQN-of-quota
  DeleteQuota.AllowFileSystemOffline=
    {true | false}
```

OPTIONS

Job

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the job to modify.

Name

Identifies the name of the scheduled job.

Schedule

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the schedule with which this job is associated.

Enabled [optional]

Identifies whether the scheduled job is enabled.

- Enable this option to perform the job at the scheduled time.
- Disable this option to hold the scheduled job until you enable it at a later time.

CreateFileSystemSnapshot.Name [optional]

Identifies the name that the Pillar Axiom storage system assigns to a scheduled data replica.

CreateFileSystemSnapshot.FS [optional]

Enter the GUID or FQN of the filesystem from which the data replica is created.

PerformUpdate.UpdatePackage [optional]

Identifies the name of an update file.

- The staging process uploads the file to the Pilot and expands the compressed .tar.gz file.
- The validation process checks the compatibility of the firmware and software in the update file against previously installed firmware and software.
- The update process installs the uncompressed package on the Pillar Axiom storage system.

Select a single update file for the given operation.

PerformUpdate.UpdateComponentName[1-5] [optional]

Identifies the name of an updatable component. Choose between one and six of the following options.

Table 21 Updatable components

Name	Description
"Pilot Software"	Software that runs on the Pilot, such as the GUI interface and web server, online help, SNMP, and NDMP
"Pilot OS"	Operating system for the Pilot
"Slammer Software"	NAS or SAN software that runs on Pillar Axiom 300 and 500 Slammers
"Slammer Software AX600"	NAS or SAN software that runs on Pillar Axiom 600 Slammers
"Slammer PROM"	Programmable ROM (PROM), that includes BIOS and netboot code, for Pillar Axiom 300 and 500 NAS or SAN Slammers
"Slammer PROM AX600"	Programmable ROM (PROM), that includes BIOS and netboot code, for Pillar Axiom 600 Slammers
"Brick Firmware"	RAID firmware for SATA Bricks
"Brick FC Firmware"	FC firmware for Bricks
"Brick SATA2 Firmware"	RAID firmware for version 2 SATA Bricks

PerformUpdateDiskDrive.UpdatePackage [optional]

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick to update.

PerformUpdateDiskDrive.
UpdateComponentName [optional]

Identifies the name of the updatable Brick.

CreateQuota.AllowFileSystemOffline [optional]

Identifies whether the Pillar Axiom system can take the filesystem offline to perform quota operations.

- Enable this option if the Pillar Axiom system can take the filesystem offline to create or remove a directory-level quota when the specified directory contains files and subdirectories.
- Disable this option if the system cannot take the filesystem offline to perform the quota operation when the specified directory contains files and subdirectories. The filesystem remains online, and the directory-level quota is not created or removed.

Whichever setting you choose, the filesystem remains online and the quota operation is performed when the specified directory is empty.

CreateQuota.FileSystem [optional]

Enter the GUID or FQN of the filesystem for which the quota is to be created.

CreateQuota.Path [optional]

Enter a full path, which starts with a forward slash (/), to the directory that is covered by this quota.

CreateQuota.RestrictUsers [optional]

Identifies who the quota covers. Choose from:

- AllAggregate if the quota covers all users who store data in the specified directory. Leave the Username and Groupname fields blank.
- SingleUser if the quota covers one user. Identify which user the quota covers in the Username field.
- Group if the quota covers a group of users. Identify which group the quota covers in the Groupname field.

CreateQuota.EnforceAllLimitsForDirectory [optional]

Identifies whether the system enforces and tracks quota limits. All quotas have automatic usage tracking. You can optionally enable enforcement for each quota.

- If RestrictUsers=AllAggregate, enable this option so that the Pillar Axiom system enforces all quota limits in the specified directory.
- If RestrictUsers=SingleUser or Group, disable this option. The system tracks and enforces the user's or group's quota limits in the filesystem.

CreateQuota.Username [optional]

Identifies a user that the quota covers. Enter a user name, user ID (UID), CIFS account name, or NFS account name. Field types support UTFString.

CreateQuota.Groupname [optional]

Identifies a group of users that the quota covers. Enter a group name or group ID (GID).

CreateQuota.SoftLimitInMB [optional]

Identifies a soft capacity limit, which means that once the system reaches this value, it sends an administrator alert. If space is available, the system stores data until it reaches the grace period or the hard limit.

Be aware that:

- If a parent directory's quota limit is smaller than a child directory's, the system enforces the parent's limit.
- Users sometimes delete directories from a filesystem's directory structure. If Directory, Directory & User, or Directory & Group quotas are assigned to the deleted directory, the quotas are also deleted.

To define an unlimited value, enter 0 (zero).

CreateQuota.HardLimitInMB [optional]

Identifies a hard capacity limit, which means that once the system reaches this value, it rejects write requests and does not store data.

To define an unlimited value, enter 0 (zero).

CreateQuota.MBGracePeriodInDays [optional]

Identifies the number of days that the system can exceed a soft limit. When the time expires, the soft limit becomes a hard limit.

To define an unlimited value, enter 0 (zero).

DeleteQuota.ID [optional]

Enter the GUID or FQN of the quota to remove.

DeleteQuota.AllowFileSystemOffline [optional]

Identifies whether the Pillar Axiom system can take the filesystem offline to perform quota operations.

- Enable this option if the Pillar Axiom system can take the filesystem offline to create or remove a directory-level quota when the specified directory contains files and subdirectories.
- Disable this option if the system cannot take the filesystem offline to perform the quota operation when the specified directory contains files and subdirectories. The filesystem remains online, and the directory-level quota is not created or removed.

Whichever setting you choose, the filesystem remains online and the quota operation is performed when the specified directory is empty.

EXAMPLES

Run the following command to modify a scheduled job that removes quota tracking and enforcement from the
/src directory in the EngAtl filesystem.

```
pdscli request ModifyJob @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyJob
Job=IDA0479DC2-38EF-9828-8902-E0FC03743270
Name=Delete-quota
Schedule=IDf1234567-89ab-cde0-0123-
  456789eeeee0
Enabled=true
DeleteQuota.ID=/Company-XYZ-FS1/EngAtl:/src
DeleteQuota.AllowFileSystemOffline=true
```

See the ModifyJobResponse.txt file for example response data.

SEE ALSO

CreateJob on p. 127

DeleteJob on p. 181

Ranges for Option Values on p. 57

NAME

ModifyLUN

DESCRIPTION

Modifies a LUN in a Pillar Axiom storage system and allows you to efficiently partition (stripe) storage across multiple Bricks based on storage or performance priorities. A LUN is defined as:

A logical volume within a storage area network (SAN). Administrators assign storage resources and Quality of Service (QoS) attributes to each logical unit (LUN).

Although you can modify the capacity of an existing LUN, the ability to use the added capacity depends entirely on the operating system and filesystem type of the SAN host that is accessing the LUN. For example, whereas some operating systems have the ability to dynamically resize a LUN, some do not allow LUN resizing. Some filesystems can scale to use the added capacity, and some cannot.

Never attempt to reduce the capacity of a LUN.

SYNTAX

```
pdscli {request | submit} ModifyLUN
  LUN=LUN-name
  VolumeGroup=GUID-or-FQN-of-volume-group
  Name=LUN-name
  AutomaticallyAssignSlammer={true | false}
  Slammer=GUID-or-FQN-of-Slammer
  ControlUnitNumber={0 | 1}
  LUNNumber=integer-0-through-255
  Mapped={true | false}
  EnableiSCSIAccess={true | false}
  EnableFibreChannelAccess={true | false}
  Capacity=integer-0-through-2147483647
  Max=integer-1-through-2147483647
  Profile={OracleASM | Normal | HighThroughput}
  RelativePriority={Low | Medium | High | Archive | Premium}
  AccessBias={None | Sequential | Random | Mixed}
  IOBias={Mixed | Read | Write | None}
  TemporaryPriority={Low | Medium | High | Archive | Premium}
  StorageClass={MultiLevelCellSolidStateDrives |
    SingleLevelCellSolidStateDrives | FibreChannelHardDiskDrives |
    SataHardDiskDrives}
  Redundancy={Standard | Double}
  SuppressConservativeMode{true | false}
  SnapLUNStorage.MaximumCapacity=integer-0-through-2147483647
  CloneLUNStorage.MaximumCapacity=integer-0-through-2147483647
  ForceSlammerAssignment={true | false}
  BackgroundCopyPriority=
    {MaximizeSpeed | MaximizeImpact | SystemChooses}
```

OPTIONS

LUN

Identifies fully qualified name (FQN) of the LUN to be modified. This parameter is required.

VolumeGroup

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the volume group with which the LUN is associated.

Name

Identifies the name that is assigned to a LUN for administrative purposes. LUN names must be unique across the Pillar Axiom system, not just within its associated volume group.

Name is a required parameter only if you want to change the FQN LUN's user-visible name.

Slammer [optional]

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer with which the LUN is associated.

Note: If you do not enter a value, the system will automatically designate a Slammer. If AutoAssignSlammer is false, the specified value changes the Slammer assignment.

ControlUnitNumber [optional]

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs. The CUs are identified by 0 (the top CU) and 1 (the bottom CU).

Note: If you do not enter 0 or 1, the system automatically designates a CU. If AutomaticallyAssignSlammer is false, the specified value changes the CU assignment.

LUNNumber [optional]

Identifies the unique number that is assigned to a LUN and can be accessed by all SAN hosts if the LUN is not mapped. If the LUN is mapped, the LUN number must be unique to the mapped SAN host.

Note: If you intend to map this LUN to a specified host, do not include the LUNNumber option.

Mapped [optional]

Identifies whether all hosts or only selected hosts may access this LUN.

- Mapped allows only designated hosts to access this LUN.
- Unmapped allows all hosts to access this LUN using the LUN Number.

Note: If you choose Mapped, do not assign a LUNNumber.

EnableiSCSIAccess

Identifies that the iSCSI protocol specification is used when accessing the newly created SAN LUN.

Note: You must specify iSCSI, Fibre Channel, or both protocols.

EnableFibreChannelAccess

Identifies that the Fibre Channel (FC) protocol specification is used when accessing the newly created SAN LUN.

Note: You must specify iSCSI, Fibre Channel, or both protocols.

Capacity

Identifies the capacity that is assigned to this LUN.

Max [optional]

Identifies the maximum capacity limit that is assigned to the object.

Profile [optional]

Identifies a performance profile. Choose from:

- Normal, which defines specific performance and redundancy settings for each logical volume. This value is highly recommended and should be used in almost all cases.
- OracleASM, which uses a wide stripe RAID geometry consisting of 1 MB strips to improve the overall performance of logical volumes in an Oracle ASM environment.

- HighThroughput, which stripes the filesystem or LUN across all RAID arrays. SATA Bricks have two RAID groups per Brick, and FC Bricks have one RAID group per Brick. You do not need to set double redundancy to get full array striping.

Important! Use of the HighThroughput profile is for special purposes only. Do not use this profile for ordinary filesystems or LUNs. Please contact the Pillar Customer Services Advanced Solutions Group before using this profile option, which should only be used for special testing where the system configuration is cleared after the test.

RelativePriority

The Relative Priority determines how much of the system resources are devoted to that volume, including the allocation of Slammer CPU cycles and the allocation of specific portions of the disk platters. The higher the Priority, the greater the allocation of CPU time and the faster the media access time.

The Relative Priority option prioritizes the layout of data that is stored in the Pillar Axiom system based on one of the following settings:

- Premium priority is the highest possible performance. This priority uses the outer 20% of the drive platters.

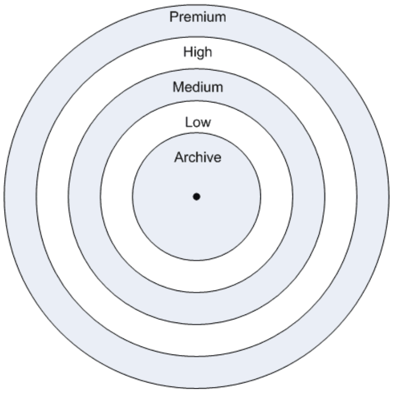
Note: I/O is typically faster when data resides on the outer edge of the drive platters.

- High priority logical volumes are allocated space in the outer 20%-40% of the drive platters.

Tip: If sufficient Bricks are present, performance of high priority volumes is enhanced by placing those volumes on a greater number of Bricks.

- Medium priority logical volumes are allocated space in the outer 40% to 60% of the drive platters.
- Low priority logical volumes are allocated space in the band that is 60% to 80% from the outer diameter of drive platters.
- Archive (lowest-priority) logical volumes are allocated space in the inner areas of the drive platters.

Figure 4 Example of HDD priority bands



If needed, the Pillar Axiom system allocates space in a higher or lower priority band (within the same Storage Class) than the one that you choose.

AccessBias

Identifies the typical data access method. Choose from:

Table 22 Data access methods

Access bias	I/O bias	Read ahead in the Slammer	RAID configuration in the Brick
Sequential	Read	Aggressive	RAID 5 Reads large extents of the data into memory.
	Write	Conservative	RAID 5 • Allocates capacity in large, full-stripe (640 K) extents. • Writes data to physical storage more quickly and in full-stripe extents.
Mixed and random	Read	None	RAID 5 • Reads small extents into memory. • Allocates capacity in small chunks. • Writes data to physical storage more slowly to combine (or eliminate) write operations.
Random	Write	None	RAID 10

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

IOBias

Identifies the typical read-write ratio. Choose from:

- Read if users or applications read data more often than they write to the data source.
- Write if users or applications write data more often than they read it.

Important! If you choose Random as the access method and Write as the I/O Bias, the system creates the volume with a Distributed RAID geometry. This geometry provides enhanced write performance but uses twice the capacity.

- Mixed if the read-write ratio varies.

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

The system stores all writes of user data and system metadata in mirrored copies of the journal. Although the filesystem or LUN is journaled, for some of the profiles, write caching is disabled because it tends to have very low hit rates.

One copy is maintained in non-volatile memory on one control unit (CU) of a Slammer. The mirror copy is maintained in one of:

- Battery-backed memory of the partner CU on the Slammer (preferred location). Writes to this copy are equivalent to write-back cache.
- An area reserved on disk for the logical volume if the partner CU is unavailable for the write. Writes to this copy are equivalent to write-through cache.

Writes from the journal to permanent disk storage are equivalent to write-through cache. The system flushes user data and the corresponding metadata as a unit to disk.

TemporaryPriority

Temporarily changes the priority level of the LUN without changing the physical location of the volume. When you want to change the priority back to the original level, modify the QoS of the volume.

One use for temporarily changing the priority level is to allow for a restore operation or data load operation where the original QoS would have been set to optimize for Write Archive or Random Read or Write, respectively.

Important! If the temporary QoS change is not changed back to the original QoS, it can have adverse effects on other system resources. If the change is to be made permanent, the QoS should be changed and the data migrated.

StorageClass

Identifies the category of physical storage on which the logical volume resides:

- Fibre Channel drives
- SATA drives

- SATA single-level cell solid state drives
- SATA multi-level cell solid state drives

Redundancy

Identifies how many mirror copies of the original data are stored online.

Important! Pillar highly recommends that you consult with a Pillar customer support professional for assistance with sizing your system and creating your volumes (filesystems and LUNs).

Redundancy options include:

- Standard, to store original data only. Data striping over multiple RAID groups maintains full redundancy, even without mirror copies.
- Double, to store original data and one mirror copy, with data striping over multiple RAID groups.

Note: Double Redundancy can only provide true redundancy if your system has enough Bricks to allocate the filesystem or LUN such that no two mirror copies share a RAID group.

SATA Bricks have two RAID groups per Brick, FC Bricks have one RAID group per brick.

If the storage pool is becoming depleted, or a large filesystem or LUN is created, it might be necessary to place the filesystem or LUN on more RAID group fragments.

Depending on the system’s ability to allocate sufficient contiguous storage blocks for the size of the logical volume, use the following number of RAID groups to configure your volumes for best performance:

Table 23 Optimum number of RAID groups for best performance

Priority	SATA standard Redundancy	SATA double redundancy	FS standard redundancy	FS double redundancy
Archive	4	8	2	4
Low	4	8	2	4
Medium	6	12	3	6
High	8	16	4	8
Premium	8	16	4	8

For *performance testing purposes only*, create a filesystem or LUN using standard redundancy and the High Throughput Profile. This is NOT recommended for most applications. Reset your system after you have created a High Throughput LUN or filesystem before you configure normal filesystems or LUNs for applications.

SuppressConservativeMode

Identifies that the LUN continues to use write-back cache even if the control unit (CU) is in a degraded mode.

Important! Before setting SuppressConservativeMode to TRUE, consult with the Pillar World Wide Customer Support Center to discuss the implications of this mode. When operating in this mode, data loss will occur when operating with a failed control unit when a power failure occurs.

SnapLUNStorage.MaximumCapacity

Identifies additional capacity used for snapshot storage that is assigned to this LUN. As a guideline, create Clone LUN space equal to or greater than the original LUN (100 to 150%). The Clone LUN does not consume space from the original LUN.

Important! If the Clone LUN reaches the maximum capacity limit, the system will notify you and automatically delete any Clone LUN that has exceeded the limit.

CloneLUNStorage.MaximumCapacity

Identifies additional capacity used for snapshot storage that is assigned to this LUN. As a guideline, create Clone LUN space equal to or greater than the original LUN (100 to 150%). The Clone LUN does not consume space from the original LUN.

Important! If the Clone LUN reaches the maximum capacity limit, the system will notify you and automatically delete any Clone LUN that has exceeded the limit.

Note: Both SnapLUNStorage.MaximumCapacity and CloneLUNStorage.MaximumCapacity parameters perform the same function. CloneLUNStorage.MaximumCapacity simply reflects the new naming conventions.

BackgroundCopyPriority

Identifies the strategy the system should use to control the impact on performance when background tasks need to copy or move data from one location in the storage pool to another:

- MinimizeImpact. Restricts the amount of work performed on a loaded system. This option is intended to have a minimal impact on client I/O throughput at the expense of longer copy times.
- SystemChooses. Balances the background copy with the incoming client I/O. This option is the default.
- MaximizeSpeed. Prioritizes the background copy at the expense of client I/O throughput.

Note: When the system is idle or lightly loaded, the above background task maximizes the amount of work done regardless of the option selected.

The following types of operations are affected by the strategy you select:

- Copy
- Restore
- Quality of Service (QoS) Priority changes
- Storage Class changes

Data transfer operations invoked by the Pillar Axiom MaxRep Replication for NAS utility are not affected by the BackgroundCopyPriority setting.

EXAMPLES

Run the following command to create a LUN for the Engineering organization.

```
pdscli request ModifyLUN @input
```

```
where input.req contains the following name/value pairs:
Request=ModifyLUN
EnableiSCSIAccess=true
EnableFCAccess=false
VolumeGroup=/Eng
LUN=/EngLUN
Name=EngLUN
LUNNumber=0
Mapped=false
Profile=Normal
Capacity=500
RelativePriority=Archive
AccessBias=Random
IOBias=Write
StorageClass=FibreChannelHardDiskDrives
Redundancy=Standard
BackgroundCopyPriority=MaximizeSpeed
```

See the ModifyLUNResponse.txt file for example response data.

SEE ALSO

CreateLUN on p. 135

DeleteLUN on p. 182

GetAllLUNs on p. 221

GetLUNDetails on p. 269

GetLUNMappingConfigDetails on p. 270

PerformConfigureLUNMapping on p. 412

PerformRebuildLUNRedundancy on p. 475

Ranges for Option Values on p. 57

NAME **ModifyManagementConfig**

DESCRIPTION Modifies the global settings that apply to the entire Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} ModifyManagementConfig
   Name=string
   IPAddress=IP-address
   Pilot[1-2]IPAddress=IP-address
   IPNetMask=IP-mask
   GatewayIPAddress=IP-address
   DHCPEnabled={true | false}
   TransmitSetting={FULL_1000 | HALF_1000 | FULL_100 | HALF_100 |
      FULL_10 | HALF_10 | AUTO}
   ServerIPAddress=IP-address
   EmailNotificationEnabled={true | false}
   CallHomeEnabled={true | false}
   PrimaryDNSServer=IP-address
   SecondaryDNSServer=IP-address
   DisallowLargeFiles={true | false}
   TransferMethod={HTTPS | SCP}
   ServerLocation={minLen | maxLen}
   ScpIPAddress={minLen | maxLen}
   Username=login-name
   Path=/path/to/directory
   ScpPassword={minLen | maxLen}
   ExternalHostTransfer={true | false}
   UseProxy={true | false}
   ProxyIP=Proxy-IP-address
   ProxyPort=integer
   ProxyType={SOCKS5 | SOCKS4 | HTTP}
   MaximumFailedLogins=integer
   SessionTimeout=duration-value
   RemoteReplicationConfig.RemoteFile
      ReplicationConfig.ManagementIPAddress=
      Remote-File-IP-address
   RemoteReplicationConfig.RemoteVolume
      ReplicationConfig.ManagementIPAddress=
      Remote-Volume-IP-Address
   iSCSIGlobalConfig.iSCSIAlias
      {Per-initiator | All Initiators}
   iSCSIGlobalConfig.Authentication=
      {Per-initiator | All Initiators}
   iSCSIGlobalConfig.EnableAccessControl=
      {true | false}
   iSCSIGlobalConfig.EnableBiDirectionalCHAP=
      {true | false}
   iSCSIGlobalConfig.CHAPPassword1=string
   iSCSIGlobalConfig.CHAPPassword2=string
   iSCSIGlobalConfig.EnableHeaderDigest=
      {true | false}
   iSCSIGlobalConfig.EnableDataDigest=
      {true | false}
   iSNSGlobalConfig.iSNSInitiatorAccessControl=
      {iSNS | Axiom | None}
   iSNSGlobalConfig.iSNSEnabled={true | false}
   iSNSGlobalConfig.iSNSDHCPDiscoveryEnabled=
      {true | false}
   iSNSGlobalConfig.iSNSServerIPAddress={IP-address}
   iSNSGlobalConfig.iSNSServerTCPPort={0 | integer}
```

OPTIONS

Name [optional]

Identifies the name that is assigned to the Pillar Axiom storage system.

IPAddress [optional]

Identifies the IP address that is permanently assigned to the Pillar Axiom system if DHCP is disabled.

PilotIP[1-2]Address [optional]

Identifies the primary and secondary Pilot control unit.

IPNetmask [optional]

Identifies the subnet mask for the Pillar Axiom IP address if DHCP is disabled.

GatewayIPAddress [optional]

Identifies the IP address that is assigned to the gateway server if DHCP is disabled.

DHCPEnabled [optional]

Identifies whether Dynamic Host Configuration Protocol (DHCP) is enabled.

- Enable DHCP if you have a DHCP server that automatically assigns IP addresses to network clients. This setting makes the Pillar Axiom system known to the DHCP software.
- Disable DHCP if you do not use it in your network.

TransmitSetting [optional]

Identifies the port speed and duplex setting for the management interface on the Pillar Axiom System.

Important! Use care when setting the transmit speed and duplex mode. A management interface setting that is not supported by the external network could result in loss of access to the Pilot. If access is lost, contact the Pillar World Wide Customer Support Center for assistance.

Note: Auto negotiate is the default transmit setting. We recommend the default setting for all but special circumstances.

ServerIPAddress [optional]

Identifies an electronic mail (email) server that receives alerts. Enter the IP address of an email server in which the recipients have email accounts.

EmailNotificationEnabled [optional]

Identifies whether email is enabled.

- Enable email if you intend to define alerts to send email notifications.
- Disable email if you do not want to send email notifications from the Pillar Axiom system.

CallHomeEnabled [optional]

Identifies whether Call-Home is enabled.

- Enable Call-Home support so that the Pillar Axiom system sends status messages to the Pillar World Wide Customer Support Center. You can create alerts to send notifications to recipients within your organization.
- Disable Call-Home support so that the Pillar Axiom system does not send the status messages.

PrimaryDNSServer AND SecondaryDNSServer [optional]

Identifies the Domain Name Sever (DNS) that is used to resolve IP addresses. This includes an email server that sends Call-Home messages from the Pillar Axiom storage system to the Pillar World Wide Customer Support Center. It can also include the Call-Home server, callhome.support.pillardata.com, that receives Call-Home messages by means of SCP or HTTPS. Call-Home messages contain information about system issues and failures.

DisallowLargeFiles [optional]

Identifies whether trace logs and performance statistics are included in Call-Home notifications.

- Disallow large files so that the Pillar Axiom system excludes trace logs and performance statistics from the Call-Home messages that are sent to the Pillar World Wide Customer Support Center. You can collect, download, and transmit the trace logs separately if they are needed.
- Allow large files so that trace logs and performance statistics are automatically included in the messages that are sent to the Pillar World Wide Customer Support Center.

TransferMethod [optional]

Specifies the method of Call-Home file transfers over the internet using:

- SCP: Uses the secure copy (SCP) method with 1024-bit encryption and secure keys.

- HTTPS: Uses the Hypertext Transfer Protocol Secure method by sending files directly to Pillar or through a proxy server for security purposes. This method can also be used when the Pillar Axiom system does not have direct access to the Internet.

ServerLocation [optional]

Specifies the name or IP address of the server to where requests are sent.

ScpIPAddress [optional]

Specifies the IP address of the server when using SCP transfers.

UserName [optional]

Identifies the administrator's login (user) name.

Path [optional]

The path on the server to which the files are transferred.

ScpPassword [optional]

Identifies the administrator's password for SCP transfers.

ExternalHostTransfer [optional]

Allows Call-Home file transfers to be sent outside of the customer network.

UseProxy [optional]

Enables or disables the use of the proxy.

ProxyIP [optional]

Specifies the IP address of the proxy.

ProxyPort [optional]

A port setting that defines which port the proxy will use to send and receive information.

ProxyType [optional]

Specifies the proxy type that is used to access the Internet: SOCKS (SOCKS4 and SOCKS5 should be used behind a firewall), or HTTP.

MaximumFailedLogins [optional]

Identifies the number of times that an administrator can attempt, but fail, to log in to the Pillar Axiom system.

SessionTimeout [optional]

Identifies an inactivity time limit, after which an administrator's session is terminated. In-progress sessions are not affected by changes that you make to the value; current sessions use the old value. Sessions that start after you change the value use the modified session timeout.

The default for the session timeout is 20 minutes. Values for the session timeout parameter must be between 1 and 999 minutes.

If the time period contains any values lower than a day, you must include the "T" separator. For example, 2 hours=PT2H; 2 hours and 30 minutes = PT2H30M, where P = period of time; T = time; H = hours, and M = minutes. Other session timeout elements include Y = years and D = days.

RemoteReplicationInterface.ManagementIPAddress [optional]

Identifies the IP address of a remote storage device.

iSCSIGlobalConfig.iSCSIAlias [optional]

Specifies the iSCSI Alias by which SAN hosts can identify the Pillar Axiom using iSCSI. The argument for this parameter (if present) should be a UTF string of up to 255 characters.

iSCSIGlobalConfig.Authentication [optional]

Indicates whether or not the Pillar Axiom will perform CHAP authentication with SAN host iSCSI initiators ("All initiators") during login, or only for specific iSCSI initiators ("Per-initiator"). If set to "All initiators," then the CHAP Name and CHAP Secret for all SAN host iSCSI initiators must be specified to the Pillar Axiom, using the ModifySANHost request. If you set the authentication to "Per-initiator," it is only necessary to specify CHAP Names and CHAP Secrets for those initiators for which authentication has been enabled.

iSCSIGlobalConfig.EnableAccessControl [optional]

If enabled, the Pillar Axiom software rejects iSCSI login attempts from SAN host iSCSI initiators that have not explicitly been granted permission by the user through the Pillar Axiom user interface.

iSCSIGlobalConfig.EnableBiDirectionalCHAP [optional]

If enabled, the Pillar Axiom will require CHAP authentication from SAN host iSCSI initiators during login. Those initiators will need to be configured on the host with the Pillar Axiom's CHAP Name and CHAP Secret. If BiDirectionalCHAP support is disabled on the Pillar Axiom, BiDirectionalCHAP must be disabled for initiators on the SAN host machine or initiator login will fail.

iSCSIGlobalConfig.CHAPPassword1 [optional]

This parameter specifies the CHAP Secret that must be provided by SAN host iSCSI initiators for CHAP authentication during login when BiDirectionalCHAP is enabled.

iSCSIGlobalConfig.CHAPPassword2 [optional]

This parameter, identical to CHAPPassword1, specifies the expected CHAP Secret. The string typed for this parameter must be the same as that typed for CHAPPassword1.

iSCSIGlobalConfig.EnableHeaderDigest [optional]

An optional component in the iSCSI structure that checks all the iSCSI header information. If enabled, the Pillar Axiom negotiates for iSCSI header digests during session login for all initiators.

iSCSIGlobalConfig.EnableDataDigest [optional]

An optional component in the iSCSI structure that provides additional error checking for the data portion of the iSCSI packet. If enabled, the Pillar Axiom storage system negotiates for iSCSI data digests during session login for all initiators.

iSNSGlobalConfig.iSNSEnabled [optional]

Specifies iSNS-based access control in which only those initiators that are members of Discovery Domains of which the Pillar Axiom storage system is a member will be allowed to login to the Pillar Axiom storage system.

iSNSGlobalConfig.iSNSDHCPDiscoveryEnabled [optional]

This option allows Pillar Axiom iSCSI targets to be registered in the iSNS server using DHCP for the discovery of iSNS server IP address.

This option requires that you use a Microsoft DHCP server that has been configured by the Microsoft iSNS Server installer to return the server IP address using DHCP option 43 (vendor-specific) or DHCP option 83 (iSNS).

Note: Microsoft does not support DHCP option 83 until the Windows Server 2008 release.

iSNSGlobalConfig.iSNSServerIPAddress [optional]

Specifies the iSNS server IP address in which to use for static addressing for the discovery of iSNS server IP address.

iSNSGlobalConfig.iSNSServerTCPPort [optional]

Specifies the iSNS server TCP port in which to use for static addressing for the discovery of iSNS server IP address.

EXAMPLES

Run the following command to modify the global settings in your system.

```
pdscli request ModifyManagementConfig @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyManagementConfig
Name=Company-XYZ-data
DHCPEnabled=true
ServerIPAddress=172.18.65.12
EmailNotificationEnabled=true
CallHomeEnabled=true
PrimaryDNSServer=192.168.1.55
DisallowLargeFiles=true
TransferMethod=scp source-filename target-filename
UserName=Nancy
```

```
ExternalHostTransfer=true
UseProxy=true
ProxyPort=0
ProxyType=HTTP
MaximumFailedLogins=3
SessionTimeout=PT2H20M
iSCSIGlobalConfig.Authentication=per-initiator
iSCSIGlobalConfig.EnableAccessControl=false
iSCSIGlobalConfig.EnableBiDirectionalCHAP=true
iSCSIGlobalConfig.CHAPPassword1=H39pe8m2
iSCSIGlobalConfig.EnableHeaderDigest=true
iSCSIGlobalConfig.EnableDataDigest=false
```

See the ModifyManagementConfigResponse.txt file for example response data.

SEE ALSO

ModifyTimeConfig on p. 386

Ranges for Option Values on p. 57

NAME **ModifyMyAccount**

DESCRIPTION Modifies the current, logged-in administrator’s account.

SYNTAX

```
pdscli {request | submit} ModifyMyAccount
   Fullname="First Last"
   Password=string
   RetypePassword=string
   EmailAddress=name@domain.ext
   PhoneNumber=telephone-number
```

OPTIONS

Fullname [optional]

Identifies the administrator’s full name (first and last).

Password

Identifies the login password. Passwords are case sensitive, and blank passwords are not permitted.

RetypePassword

Enter the password again.

EmailAddress [optional]

Identifies the recipient’s email address. The email server to which the Pillar Axiom system sends alerts must be able to send messages to this email address.

PhoneNumber [optional]

Identifies the recipient’s telephone number. The Pillar Axiom system does not verify the validity of this entry.

EXAMPLES Run the following command to modify your login password.

```
pdscli request ModifyMyAccount @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyMyAccount
Password=H39pe8mZ
RetypePassword=H39pe8mZ
```

See the ModifyMyAccountResponse.txt file for example response data.

SEE ALSO CreateAccount on p. 92

DeleteAccount on p. 169

ModifyAccount on p. 311

Ranges for Option Values on p. 57

NAME

ModifyNASConfig

DESCRIPTION Modifies the system’s NAS configuration.

SYNTAX

```
pdscli {request | submit} ModifyNASConfig
   LinkAggregation={true | false}
   EnableAutomaticControlUnitRecovery=
      {true | false}
```

OPTIONS

LinkAggregation

Identifies whether link aggregation is enabled. Pillar Axiom systems support the IEEE 802.3ad standard.

- Enable link aggregation to treat multiple physical links as one logical link. Configure virtual network interfaces on port 0.
- Disable link aggregation if you do not need to use link aggregation. Configure virtual network interfaces on both ports.

EnableAutomaticControlUnitRecovery

Identifies whether the Pillar Axiom storage system should perform an automatic recovery operation when a previously unavailable Slammer control unit (CU) becomes available.

- Enable this option if interruptions of up to 30 seconds in data-path connections are permitted while the automatic recovery takes place.
- Disable this option if you want to be prompted to perform the operation manually when the Slammer CU is ready for recovery.

EXAMPLES Run the following command to modify the NAS configuration.

```
pdscli request ModifyNASConfig @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyNASConfig
LinkAggregation=true
EnableAutomaticControlUnitRecovery=true
```

See the ModifyNASConfigResponse.txt file for example response data.

SEE ALSO GetNASConfigDetails on p. 274

ModifyManagementConfig on p. 360

Ranges for Option Values on p. 57

NAME **ModifyNDMPConfig**

DESCRIPTION Modifies the Netword Data Management Protocol (NDMP) configuration settings.

SYNTAX

```
pdscli {request | submit} ModifyNDMPConfig
   Enabled={true | false}
   PortNumber=integer
   Username=string
   FileServer=GUID-or-FQN-of-File-Server
   Password=string
   RetypePassword=string
```

OPTIONS

Enabled [optional]

Identifies whether Network Data Management Protocol (NDMP) support is enabled.

- Enable NDMP to use NDMP-compliant backup applications to back up Pillar Axiom data.
- Disable NDMP to inactivate support for NDMP-compliant backup applications.

PortNumber [optional]

Identifies a TCP port that is assigned as the listen port for the NDMP daemon. The default NDMP port number is 10000.

Username [optional]

Identifies the NDMP user's login name.

FileServer [optional]

Identifies the File Server with which the NDMP attributes are associated. Although you enable or disable the NDMP feature system-wide, all NDMP backup and restore operations must go through this one File Server.

Enter the GUID or FQN of the associated File Server.

Password [optional]

Identifies the NDMP user's password.

RetypePassword [optional]

Enter the password again.

EXAMPLES Run the following command to modify the NDMP configuration in your system.

```
pdscli request ModifyNDMPConfig @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyNDMPConfig
Enabled=true
PortNumber=10000
Username=NDMP4pillar
```

```
FileServer=/Company-XYZ-data/Company-XYZ-FS1
Password=N2p8
RetypePassword=N2p8
```

See the ModifyNDMPConfigResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **ModifyNFSHostEntry**

DESCRIPTION Modifies the specified Network File system (NFS) host.

SYNTAX

```
pdscli {request | submit} ModifyNFSHostEntry
   NFSHostEntry=GUID-or-FQN-of-host-entry
   AllowRootAccess={true | false}
   WriteSupported={true | false}
   AuthenticationMode={System | None}
   OrderNumber=integer
```

OPTIONS

NFSHostEntry

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the NFS host entry to modify.

AllowRootAccess [optional]

Identifies whether users who are logged in to a client workstation as root are root users on the export.

- Enable this option so that root users on clients are root users. This is comparable to no_root_squash in Linux root-squashing options.
- Disable this option so that root users on clients are identified as the user nobody. This is comparable to the root_squash option.

WriteSupported [optional]

Identifies whether the export is made available to users in read-only or read-write mode.

- Enable this option so that the export operates in read-only mode.
- Disable this option so that the export operates in read-write mode.

AuthenticationMode [optional]

Identifies how users are authenticated.

- System (AUTH_SYS) means that user IDs (UIDs), group IDs (GIDs), and supplemental group IDs are passed with each NFS request, without being authenticated by the File Server.
- None (AUTH_NONE) means that users are mapped to the anonymous user “nobody.” UIDs and GIDs are not passed with the NFS requests.

We recommend that you set this option to System.

OrderNumber [optional]

Identifies the authentication order for this host relative to other hosts on the export. For this zero-based index, 0 identifies the first host that is authenticated.

We recommend that you set this option to 2.

EXAMPLES

Run the following command to change the order in which a specific host entry is authenticated.

```
pdscli request ModifyNFSHostEntry @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyNFSHostEntry
NFSHostEntry=IDA5479DC2-38EF-9828-8902-
  E0FC03743270
AuthenticationMode=System
OrderNumber=2
```

See the ModifyNFSHostEntryResponse.txt file for example response data.

SEE ALSO

CreateNFSHostEntry on p. 148

DeleteNFSHostEntry on p. 184

Ranges for Option Values on p. 57

NAME **ModifyNotificationDefinition**

DESCRIPTION Modifies the specified notification.

SYNTAX

```
pdscli {request | submit}
   ModifyNotificationDefinition
      NotificationDefinition=GUID-or-FQN-of-
         notification
      Name=notification-name
      Description=notification-description
      Client={SNMP | Email}
      EchoDefinitionInNotification=
         {true | false}
      SeverityFilter.Severity[1-10]={Critical
         | Error | Warning | Informational}
      EmailAddress[1-4]=username@domain.ext
```

OPTIONS

NotificationDefinition

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the notification to modify.

Name [optional]

Identifies the name of a notification that the Pillar Axiom system generates when an event or alert is triggered.

Description [optional]

Describes the notification in a way that is meaningful to the recipients.

Client [optional]

Identifies the notification client. Choose from:

- SNMP, if the client is a Simple Network Management Protocol (SNMP) application.
- Email, if the client is an electronic mail (email) server that sends the notifications to designated email addresses.

EchoDefinitionInNotification [optional]

Identifies whether the notification definition is included with the event details in the message that the Pillar Axiom system sends to designated recipients.

SeverityFilter.Severity[1-10] [optional]

Identifies a category of event severities. Choose from:

- Critical, which means that access to data is compromised (for example, two disks in a RAID array have failed) or data loss has occurred.
- Error, which means that administrator action is required to prevent a “hard” error (for example, a single disk in a RAID array has failed) from becoming a critical event.
- Warning, which means that administrator action is required to prevent a “soft” error from becoming an error or critical event.
- Informational, which means that a configuration change has been detected or another non-error event has occurred.

EmailAddress[1-4] [optional]

Identifies the recipient’s email address. The email server to which the Pillar Axiom system sends alerts must be able to send messages to this email address.

EXAMPLES

Run the following command to modify a notification so that email messages are sent only to Xavier. Previously, the email messages were sent to Xavier and to Yvette Smith.

```
pdscli request ModifyNotificationDefinition @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyNotificationDefinition
NotificationDefinition=IDA0479DC2-38EF-9828-
  8902-E0FC03743270
Name=Critical
SeverityFilter.Severity[1]=Critical
EmailAddress[1]=xavier@xyz.com
```

See the ModifyNotificationDefinitionResponse.txt file for example response data.

SEE ALSO

CreateNotificationDefinition on p. 152

DeleteNotificationDefinition on p. 185

Ranges for Option Values on p. 57

NAME **ModifyQuota**

DESCRIPTION Modifies the specified filesystem quota.

SYNTAX

```
pdscli {request | submit} ModifyQuota
   ID=GUID-or-FQN-of-quota
   EnforceAllLimitsForDirectory=
      {true | false}
   SoftLimitInMB={0-for-unlimited | integer}
   HardLimitInMB={0-for-unlimited | integer}
   MBGracePeriodInDays={0-for-unlimited |
      integer}
```

OPTIONS

ID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the quota to modify.

EnforceAllLimitsForDirectory [optional]

Identifies whether the quota limits are enforced as well as tracked. All quotas have automatic usage tracking. Enforcement is optional and must be enabled explicitly and separately for each quota.

- Enable this option so that the Pillar Axiom system enforces the capacity limits.
- Disable this option so that the system does not enforce the capacity limits.

SoftLimitInMB [optional]

Identifies a soft capacity limit, which means that once the system reaches this value, it sends an administrator alert. If space is available, the system stores data until it reaches the grace period or the hard limit.

Be aware that:

- If a parent directory's quota limit is smaller than a child directory's, the system enforces the parent's limit.
- Users sometimes delete directories from a filesystem's directory structure. If Directory, Directory & User, or Directory & Group quotas are assigned to the deleted directory, the quotas are also deleted.

To define an unlimited value, enter 0 (zero).

HardLimitInMB [optional]

Identifies a hard capacity limit, which means that once the system reaches this value, it rejects write requests and does not store data.

To define an unlimited value, enter 0 (zero).

MBGracePeriodInDays [optional]

Identifies the number of days that the system can exceed a soft limit. When the time expires, the soft limit becomes a hard limit.

To define an unlimited value, enter 0 (zero).

EXAMPLES

Run the following command to increase the quota on the engineering organization’s filesystem.

```
pdscli request ModifyQuota @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyQuota
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
EnforceAllLimitsForDirectory=true
SoftLimitInMB=75
HardLimitInMB=90
MBGracePeriodInDays=5
```

See the ModifyQuotaResponse.txt file for example response data.

SEE ALSO

CreateQuota on p. 154

DeleteQuota on p. 186

Ranges for Option Values on p. 57

NAME **ModifySANHost**

DESCRIPTION Modifies the specified SAN host.

SYNTAX

```
pdscli {request | submit} ModifySANHost
   SANHostName=Host-name
   HP-UXAddressing={true | false}
   LUNSettings.LUN=GUID-or-FQN-of-LUN
   LUNSettings.LoadBalancingType={Static
      | Round Robin}
   iSCSIConfig.EnableAccessControl={true | false}
   iSCSIConfig.EnableAuthentication={true | false}
   iSCSIConfig.CHAPName=CHAP-name
   iSCSIConfig.CHAPPassword1=string
   iSCSIConfig.CHAPPassword2=string
```

OPTIONS

SANHostName

Identifies the name assigned to a specific SAN host.

HPUXAddressing

Use this option when the SAN hosts that access the LUNs have HP-UX initiator ports and HP HBAs.

- If set to true, LUN numbers are determined using the HP/UX addressing scheme, allowing up to 255 LUNs.
- If set to false, the default numbering scheme is used, allowing 256 LUNs.

You can verify the current host mappings using the LUN Connections tab. The host cannot be configured with the HP-UX LUN addressing option until its mapping for LUN 0 is either deleted or converted to a mapped LUN with no access by the host.

LUNSettings.LUN

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the LUN.

LUNSettings.LoadBalancingType

Identifies the type of load balancing that the SAN hosts should perform to access LUNs that are configured on the Pillar Axiom storage system. Choose from:

- Static load balancing across multiple paths to the configured LUNs.
- Round-robin load balancing across multiple paths to the configured LUNs.

iSCSIConfig.EnableAccessControl

If Access Control has been enabled (set to true) using the ModifyManagementConfig request, this parameter specifies that the Pillar Axiom will allow access from all initiators in the SAN host.

iSCSIConfig.EnableAuthentication

If the Pillar Axiom's management configuration has been set for authentication on a "Per-Initiator" basis, this parameter specifies whether or not the Pillar Axiom will attempt CHAP authentication to all initiators in the SAN host during login. If, however, the Pillar Axiom has been configured to require authentication from "All Initiators" this parameter is ignored and CHAP authentication is always attempted.

Note: If the initiator on the SAN host has been configured to require CHAP authentication, then login will fail unless either the Pillar Axiom's management configuration has been set to authenticate to "All Initiators" or it has been set to "Per-Initiator" and EnableAuthentication is set to true. In either case, you must specify the CHAPName and CHAPSecret for the initiator.

iSCSIConfig.CHAPName

Specifies the Challenge Handshake Authentication Protocol (CHAP) name that the Pillar Axiom will use for authentication to the SAN host iSCSI initiator during login. This must be the same CHAPName with which the initiator was configured on the SAN host.

iSCSIConfig.CHAPPassword1

Specifies the CHAP Secret that the Pillar Axiom will use for authentication to the SAN host iSCSI initiator during login. This must be the same CHAP Secret with which the initiator was configured on the SAN host.

iSCSIConfig.CHAPPassword2

The second password used for CHAP authentication, identical to CHAPPassword1. The string typed for this parameter must be the same as that typed for CHAPPassword1.

EXAMPLES

Run the following command to modify the SANTest_02 host.

```
pdscli request ModifySANHost @input
```

where input.req contains the following name/value pairs:

```
Request=ModifySANHost
```

```
SANHostName=SANTest_02
HPUXAddressing=false
LUNSettings.LUN=/EngLUN
LUNSettings.LoadBalancingType=Static
iSCSIConfig.EnableAccessControl=
  false
iSCSIConfig.EnableAuthentication=
  true
iSCSIConfig.CHAPName=Company-XYZ-data
iSCSIConfig.CHAPPassword1=H39pe8m2
iSCSIConfig.CHAPPassword2=H39pe8m2
```

See the ModifySANHostResponse.txt file for example response data.

SEE ALSO

GetSANConfigDetails on p. 284

GetHostMappingConfigDetails on p. 265

GetAllSANHosts on p. 227

GetHBAPortConnections on p. 262

PerformAssociateInitiatorsToHost on p. 398

DeleteSANHost on p. 189

GetSANHostDetails on p. 285

Ranges for Option Values on p. 57

NAME **ModifySlammer**

DESCRIPTION Modifies the specified Slammer.

SYNTAX

```
pdscli {request | submit} ModifySlammer
   Slammer=GUID-or-FQN-of-Slammer
   Name=string
```

OPTIONS

Slammer

By default, Slammers are assigned names such as /Slammer1 or /Slammer2, based on their internal WUNAME. You can use this option to modify the Slammer name into something more meaningful; for example, /NAS1 or /SAN-Slammer. Use the GetAllSlammers request to obtain the existing FQN and GUID.

Name

Identifies the name that is assigned to a hardware component. Assign unique, meaningful component names to help you more easily locate specific components. The Pillar Axiom system maps the assigned name to the component's serial number and updates the map if you modify the component name.

By default, the Bricks are assigned names such as /Brick001, based on a simple sort of the component's internal Fibre Channel WUNAME. This is the logical Brick name and is not necessarily the same as the physical Brick location. You can assign any name to a Brick.

EXAMPLES Run the following command to assign a new name to a Slammer in your system.

```
pdscli request ModifySlammer @input
```

where input.req contains the following name/value pairs:

```
Request=ModifySlammer
Slammer=SC1
Name=Slammer1
```

See the ModifySlammerResponse.txt file for example response data.

SEE ALSO

GetAllSlammers on p. 231

GetSlammerDetails on p. 295

Ranges for Option Values on p. 57

NAME

ModifySnapLUN

DESCRIPTION

Modifies a Snap LUN, which is defined as:

A point-in-time, read-write copy of a LUN that you can immediately use. Snap LUNs requires no cache and minimal disk storage space.

Snap LUNs point to the original data, and have the same QoS parameters as the source LUN.

Note: Formerly called a Snap LUN, the GUI now refers to this term as Clone LUN, whereas Snap LUN is still used in the command line.

SYNTAX

```
pdscli {request | submit} ModifySnapLUN
   SnapLUN=GUID-or-FQN-of-LUN
   Name=LUN-name
   LUNNumber=integer-0-through-255
   Active={true | false}
   Mapped={true | false}
   EnableiSCSIAccess={true | false}
   EnableFibreChannelAccess={true | false}
```

OPTIONS

SnapLUN

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the LUN you wish to modify.

Name [optional]

Identifies the name that is assigned to a LUN for administrative purposes. LUN names must be unique across the Pillar Axiom system, not just within its associated volume group.

Name is a required parameter only if you want to change the FQN LUN's user-visible name.

LUNNumber [optional]

Identifies the unique number that is assigned to a LUN and can be accessed by all SAN hosts if the LUN is not mapped. If the LUN is mapped, the LUN number must be unique to the mapped SAN host.

Active [optional]

Identifies whether the Clone LUN is available to the host (active) or is not seen by the host (inactive) when it is created. Inactive Clone LUNs can be changed to active; however, once the Clone LUN is set to active, you cannot change it to inactive. Default is active.

Mapped [optional]

Identifies whether all hosts or only selected hosts may access this LUN.

- Mapped allows only designated hosts to access this LUN.
- Unmapped allows all hosts to access this LUN using the LUN Number.

Note: If you choose Mapped, do not assign a LUNNumber.

EnableiSCSIAccess

Identifies that the iSCSI protocol specification is used when accessing the newly created SAN LUN.

Note: You must specify iSCSI, Fibre Channel, or both protocols.

EnableFibreChannelAccess

Identifies that the Fibre Channel (FC) protocol specification is used when accessing the newly created SAN LUN.

Note: You must specify iSCSI, Fibre Channel, or both protocols.

EXAMPLES

Run the following command to modify the name and LUN number of SLUN_01 for the Engineering organization.

```
pdscli request ModifySnapLUN @input
```

```
where input.req contains the following name/value pairs:
Request=ModifySnapLUN
SnapLUN=/SLUN_01
Name=EngLUN
LUNNumber=0
```

See the ModifySnapLUNResponse.txt file for example response data.

SEE ALSO

DeleteSnapLUN on p. 191

GetAllSnapLUNs on p. 233

GetAllSnapLUNHierarchies on p. 232

GetSnapLUNDetails on p. 297

PerformLUNDuplication on p. 447

PerformBackgroundLUNDuplication on p. 402

Ranges for Option Values on p. 57

NAME **ModifySNMPConfig**

DESCRIPTION Modifies the Simple Network Management Protocol (SNMP) configuration settings.

SYNTAX

```
pdscli {request | submit} ModifySNMPConfig
   SNMPHost[1-65535].SNMPHostUsed=
      {true | false}
   SNMPHost[1-65535].SNMPHostIP=IP-address
   SNMPHost[1-65535].SNMPHostPort=integer
   SNMPHost[1-65535].
      SNMPCommunityString=string
```

OPTIONS

SNMPHost[1-65535].SNMPHostUsed

Identifies whether the SNMP trap host is enabled.

- Enable a trap host so that it is sent SNMP traps that the Pillar Axiom system generates.
- Disable a trap host so that it is not sent traps.

SNMPHost[1-65535].SNMPHostIP

Identifies the IP address of a workstation with an SNMP-based management application installed. This workstation receives the SNMP traps that the Pillar Axiom system generates.

SNMPHost[1-65535].SNMPHostPort

Identifies the port on which an SNMP trap host listens for the SNMP traps that the Pillar Axiom storage system generates.

SNMPHost[1-65535].SNMPCommunityString

Identifies a community for which a specific trap host should receive traps that the Pillar Axiom system generates. You can specify different community strings for each trap host so that multiple administrators can receive specific types of SNMP traps.

Default: public (note the lower-case spelling)

EXAMPLES Run the following command to configure another SNMP trap host.

```
pdscli request ModifySNMPConfig @input
```

where input.req contains the following name/value pairs:

```
Request=ModifySNMPConfig
SNMPHost[1].SNMPHostUsed=true
SNMPHost[1].SNMPHostIP=192.168.1.59
SNMPHost[1].SNMPHostPort=162
SNMPHost[1].SNMPCommunityString=public
```

See the ModifySNMPConfigResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **ModifyTimeConfig**

DESCRIPTION Modifies the manual time settings that are configured on the Pillar Axiom storage system. Run this configuration request if you do not synchronize with Network Time Protocol (NTP) servers and you want to change the factory-configured system time settings.

Note: If you are running the CIFS protocol in a Windows 2000 Active Domain, NTP is mandatory. NTP is strongly advised for NFS and may be required by some applications that use NFS.

SYNTAX

```
pdscli {request | submit} ModifyTimeConfig
   NTPServiceEnabled={true | false}
   PreferredNTPServerIPAddress=IP-address
   AlternateNTPServer[1-2]IPAddress=
      IP-address
   TimeZoneName=Time-Zone-Name
   DateTime=YYYY-MM-DDTHH:mm:SS.xx+-HH:mm
```

OPTIONS

NTPServiceEnabled [optional]

Enables NTP service.

- If NTPServiceEnabled=true, you must enter the preferred NTP server IP address.

PreferredNTPServerIPAddress [optional]

Identifies the preferred NTP server.

AlternateNTPServer[1-2]IPAddress [optional]

Identifies alternate NTP servers.

TimeZoneName [optional]

Identifies the time zone of the site at which the Pillar Axiom system is installed. Choose from:

- Abu Dhabi, Muscat
- Alaska (US)
- Amsterdam, Belgrade, Berlin, Bern, Brussels, Paris, Rome, Stockholm, Vienna
- Astana, Dhaka
- Atlantic Time (Canada), Caracas
- Auckland, Wellington, Fiji
- Azores
- Bangkok, Hanoi, Jakarta
- Beijing, Hong Kong, Perth, Singapore, Tapei
- Brisbane, Guam, Melbourne, Sydney
- Bucharest, Cairo, Helsinki, Instanbul

- Central Time (US,Canada), Mexico City, Saskatchewan
- Eastern Time (US,Canada), Bogota, Lima
- Greenland
- Greenwich Mean Time: Casablanca, Dublin, Edingburgh, Lisbon, London
- Hawaii (US)
- International Date Line West
- Islamabad, Karachi
- Magadan, Solomon Islands, New Caledonia
- Mid-Atlantic
- Midway Island, Samoa
- Moscow
- Mountain Time (US,Canada)
- Pacific Time (US,Canada),Tijuana
- Tokyo, Seoul

DateTime=YYYY-MM-DDTHH:mm:SS.xx+-HH:mm [optional]

Identifies the date and clock time for manual time settings in the Pillar Axiom system. Use the following format for the entry:

YYYY-MM-DDTHH:mm:SS.xx+-HH:mm

where:

- YYYY-MM-DD designates a four-digit year, two-digit month, and two-digit day for the date.
- T is a separator that designates the start of the time portion of the string.
- HH:mm:SS designates hours, minutes, and seconds in values for a 24-hour clock.
- xx designates a fraction of a second, to two decimal places.
- +-HH:mm designates the time zone as an offset from Coordinated Universal Time (UTC) in hours and minutes. Include the + or - prefix as appropriate for your time zone in relation to UTC.

EXAMPLES

Run the following command to set the time on your system.

```
pdscli request ModifyTimeConfig @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyTimeConfig
NTPServiceEnabled=true
PreferredNTPServerIPAddress=172.20.20.2
AlternateNTPServer1IPAddress=172.20.20.3
TimeZoneName=US/Pacific
DateTime=2004-03-19T13:59:44-08:00
```

See the ModifyTimeConfigResponse.txt file for example response data.

SEE ALSO GetCurrentDateTimeDetails on p. 247

GetCurrentTimeZoneDetails on p. 248

Ranges for Option Values on p. 57

NAME **ModifyVolumeGroup**

DESCRIPTION Modifies the specified volume group.

SYNTAX

```
pdscli {request | submit} ModifyVolumeGroup
   VolumeGroup=GUID-or-FQN-of-volume-group
   Name=string
   ParentVolumeGroup=GUID-or-FQN-of-parent-
      volume-group
   MaximumCapacity=integer
```

OPTIONS

VolumeGroup

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name.

Enter the GUID or FQN of the volume group to modify.

Name [optional]

Identifies the name that is assigned to a volume group. Each volume group name must be unique within its parent volume group.

ParentVolumeGroup [optional]

Enter the GUID or FQN of the next-highest volume group in the nested chain. If volume groups are not nested, or if the next-highest level is the System itself, enter the GUID or FQN of the System.

MaximumCapacity [optional]

Identifies the maximum capacity for the volume group. The maximum capacity of the filesystems and nested volume groups that are associated with the volume group cannot exceed this value.

A value of 0 (zero) identifies that the volume group is configured with unlimited capacity. You can increase the maximum capacity of associated filesystems and nested volume groups without constraints.

EXAMPLES Run the following command to assign a new name to a volume group.

```
pdscli request ModifyVolumeGroup @input
```

where input.req contains the following name/value pairs:

```
Request=ModifyVolumeGroup
VolumeGroup=/Eng
Name=Engineering
MaximumCapacity=0
```

See the ModifyVolumeGroupResponse.txt file for example response data.

SEE ALSO

CreateVolumeGroup on p. 167

DeleteVolumeGroup on p. 195

Ranges for Option Values on p. 57

NAME **PerformAcceptAddBrick**

DESCRIPTION Accepts a newly added Brick.

SYNTAX

```
pdscli {request | submit} PerformAcceptAddBrick
  Brick=GUID-or-FQN-of-Brick
```

OPTIONS

Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the new Brick.

EXAMPLES Run the following command to accept a replacement Brick that was installed on your system.

```
pdscli request PerformAcceptAddBrick @input
```

where input.req contains the following name/value pairs:

```
Request=PerformAcceptAddBrick
Brick=/Company-XYZ-data/Brick3
```

See the PerformAcceptAddBrickResponse.txt file for example response data.

SEE ALSO PerformIdentifyBrick on p. 433

PerformResumeBrick on p. 486

Ranges for Option Values on p. 57

Service Guide

NAME **PerformAcceptAddSlammer**

DESCRIPTION Accepts a newly added Slammer.

SYNTAX

```
pdscli {request | submit} PerformAcceptAddSlammer
   Slammer=GUID-or-FQN-of-Slammer
```

OPTIONS **Slammer**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the new Slammer.

EXAMPLES Run the following command to accept a replacement Slammer that was installed on your system.

```
pdscli request PerformAcceptAddSlammer @input
```

where input.req contains the following name/value pairs:

```
Request=PerformAcceptAddSlammer
Slammer=/Company-XYZ-data/Slammer3
```

See the PerformAcceptAddSlammerResponse.txt file for example response data.

SEE ALSO PerformIdentifySlammer on p. 438

PerformResumeSlammer on p. 489

Ranges for Option Values on p. 57

Service Guide

NAME **PerformAcceptBrickDiskDrive**

DESCRIPTION Accepts a disk drive that is installed on a Brick. A new disk drive is accepted on a Brick when the system:

- Discovers the new disk drive.
- Initializes the disk drive (which destroys all configuration and user data on the disk drive.
- Stores the disk drive's serial number permanently.

SYNTAX

```
pdscli PerformAcceptBrickDiskDrive
   BrickWWN=Brick
   DiskDriveNumber=integer
```

OPTIONS

Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick on which the new disk drive was installed.

DiskDriveNumber

Identifies a disk drive that is installed on a Brick.

EXAMPLES Run the following command to accept a replacement disk drive that was installed on Brick1.

```
pdscli request PerformAcceptBrickDiskDrive @input
```

where input.req contains the following name/value pairs:

```
Request=PerformAcceptBrickDiskDrive
Brick=/Company-XYZ-data/Brick1
DiskDriveNumber=5
```

See the PerformAcceptBrickDiskDriveResponse.txt file for example response data.

SEE ALSO GetAllBricks on p. 204

PerformAcceptAddBrick on p. 391

PerformCopybackBrickLUNs on p. 415

PerformPrepareReplaceBrickFRU on p. 461

NAME

PerformAcceptForeignDiskDrive

DESCRIPTION

Accepts a disk drive that is installed on a Brick. A disk drive is foreign when the Pillar Axiom storage system does not recognize the disk drive's serial number. Any new disk drive is considered to be foreign until the system:

- Discovers the new disk drive.
- Initializes the disk drive (which destroys all configuration and user data on the disk drive).
- Stores the Pillar Axiom system's serial number on the new Brick.

SYNTAX

```
pdscli PerformAcceptForeignDiskDrive
   Brick=GUID-or-FQN-of-Brick
   DiskDriveNumber=integer
```

OPTIONS

Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick on which the new disk drive was installed.

DiskDriveNumber

Identifies a disk drive that is installed on a Brick.

EXAMPLES

Run the following command to accept a replacement disk drive that was installed on Brick1.

```
pdscli request PerformAcceptForeignDiskDrive @input
```

where input.req contains the following name/value pairs:

```
Request=PerformAcceptForeignDiskDrive
Brick=/Company-XYZ-data/Brick1
DiskDriveNumber=5
```

See the PerformAcceptForeignDiskDriveResponse.txt file for example response data.

SEE ALSO

PerformIdentifyBrickFRU on p. 435

PerformPrepareReplaceBrickFRU on p. 461

PerformResumeBrickFRU on p. 487

Ranges for Option Values on p. 57

Service Guide

NAME **PerformAssignCIFSLocalGroups**

DESCRIPTION Assigns the Common Internet File System (CIFS) protocol to a local security group.

SYNTAX

```
pdscli {request | submit} PerformAssignCIFSLocalGroups
   FileServer=GUID-or-FQN-of-File-Server
   LocalAdministrationGroupName=GroupNameString
   LocalBackupGroupName=GroupNameString
```

OPTIONS The

FileServer

Identifies the File Server.

LocalAdministrationGroupName

Identifies the local administration group to be used for CIFS protocol for a Pillar Axiom storage system.

LocalBackupGroupName

Identifies the backup group to be used for CIFS.

EXAMPLES Run the following command to associate CIFS with a local group.

```
pdscli request PerformAssignCIFSLocalGroups@input
```

where input.req contains the following name-value pairs:

```
Request=PerformAssignCIFSLocalGroups
FileServer=ID12345678-9abc-def0-1234-56789abcdef0
LocalAdministrationGroupName=EngrCIFSAdmin
LocalBackupGroupName=EngrCIFS-BU
```

See the PerformAssignCIFSLocalGroups.txt file for example response data.

SEE ALSO CreateCIFSShare on p. 95

DeleteCIFSShare on p. 171

GetAllCIFSShares on p. 206

GetCIFSShareDetails on p. 244

ModifyCIFSShare on p. 315

NAME **PerformAssociateInitiatorsToHost**

DESCRIPTION Associates the FC ports or iSCSI initiator names to the host.

SYNTAX

```
pdscli {request | submit} PerformAssociateInitiatorsToHost
   HBAPortWWN=World-wide-number
   InitiatorName
   HBAPortName=HBA-port-name
   SANHostName=Host-name
```

OPTIONS The HBAPortWWN and the InitiatorName parameters are both optional; however, only one or the other of these two can be selected. An Invalid Parameter error is returned if both are provided.

HBAPortWWN

Identifies the address, or world wide name (WWN), of the specified host bus adapter (HBA).

InitiatorName [optional]

Identifies the iSCSI initiator to the host.

HBAPortName [optional]

Identifies the name given to the gigabit interface converter (GBIC) or port.

SANHostName [optional]

Identifies the name assigned to a specific SAN host.

EXAMPLES Run the following command to associate an HBA with a SAN host.

```
pdscli request PerformAssociateInitiatorsToHost@input
```

where input.req contains the following name/value pairs:

```
Request=PerformAssociateInitiatorsToHost
HBAPortWWN=21:00:00:e0:8b:11:b5:4a
HBAPortName=HBA 3 Port 0
SANHostName=SANTest_02
```

See the PerformAssociateInitiatorsToHostResponse.txt file for example response data.

SEE ALSO GetSANConfigDetails on p. 284

GetHostMappingConfigDetails on p. 265

GetAllSANHosts on p. 227

GetSANHostDetails on p. 285

ModifySANHost on p. 377

GetHBAPortConnections on p. 262

DeleteSANHost on p. 189

NAME **PerformBackgroundFileSystemDuplication**

DESCRIPTION Creates a duplicate of a filesystem in the background. A filesystem duplicate is defined as:

> A block-level, duplicate copy (image copy) of a filesystem. Use this data redundancy feature (called Copy Filesystem in the Pillar Axiom Storage Services Manager) to create a bit-for-bit secondary duplicate of a source filesystem for immediate use as a read-write copy. Note that the Quality of Service (QoS) may differ from the original.
>
> A duplicate copy requires greater system and storage resources than a Snap FS. To create an archival copy, create an inactive clone.

Use this request, for example, to replicate a data set for testing or to set up a new environment. User data, NFS exports, and CIFS shares are duplicated. You can change the capacity and performance settings of the duplicate filesystem.

Note: Background copies may impact storage performance and resources available to the user community during peak operating hours.

SYNTAX

```
pdscli {request | submit}
  PerformBackgroundFileSystemDuplication
      FileSystem=GUID-or-FQN-of-filesystem
```

OPTIONS **FileSystem**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem to duplicate.

EXAMPLES Run the following command to duplicate the EngAtl filesystem and run as a background process.

```
pdscli request PerformBackgroundFileSystemDuplication @input
```

where input.req contains the following name/value pairs:

```
Request=PerformBackgroundFileSystemDuplication
FileSystem=/Company-XYZ-FS1/EngAtl
```

See the PerformBackgroundFileSystemDuplicationResponse.txt file for example response data.

SEE ALSO

PerformFileSystemDuplication on p. 425

CreateFileSystemSnapshotSchedule on p. 124

DeleteFileSystemSnapshot on p. 177

DeleteFileSystemSnapshotSchedule on p. 179

GetAllFileSystemSnapshots on p. 216

GetAllFileSystemSnapshotSchedules on p. 218

GetFileSystemSnapshotDetails on p. 256

GetFileSystemSnapshotScheduleDetails on p. 258

ModifyFileSystemSnapshot on p. 340

PerformRestoreFileSystemFromSnapshot on p. 485

PerformRebuildFileSystemRedundancy on p. 474

Ranges for Option Values on p. 57

NAME **PerformBackgroundLUNDuplication**

DESCRIPTION Copies data from the source LUN to the duplicate. A LUN duplicate is defined as:

> A block-level, image copy of a source LUN used for immediate read-write access. A LUN duplicate (called Copy LUN in the Pillar Axiom Storage Services Manager) creates a reliable, bit-for-bit secondary copy of the entire allocated storage space of the source LUN.
>
> This feature requires greater system and storage resources than the Snap LUN feature. To create an archival copy, create an inactive clone.

Use this request, for example, to replicate a data set for testing or to set up a new environment. You can change the capacity and performance settings of the duplicate LUN.

Note: Background copies may impact storage performance and resources available to the user community during peak operating hours.

SYNTAX

```
pdscli {request | submit}
  PerformBackgroundLUNDuplication
    LUN=GUID-or-FQN-of-LUN
  EnableiSCSIAccess={true | false}
  EnableFibreChannelAccess={true | false}
```

OPTIONS

LUN

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the LUN to duplicate.

EnableiSCSIAccess

Identifies that the iSCSI protocol specification is used when accessing the newly created SAN LUN.

Note: You must specify iSCSI, Fibre Channel, or both protocols.

EnableFibreChannelAccess

Identifies that the Fibre Channel (FC) protocol specification is used when accessing the newly created SAN LUN.

Note: You must specify iSCSI, Fibre Channel, or both protocols.

EXAMPLES

Run the following command to duplicate EngLUN and run as a background process.

```
pdscli request PerformBackgroundLUNDuplication
  @input
```

where input.req contains the following name/value pairs:

```
Request=PerformBackgroundLUNDuplication
LUN=/EngLUN
EnableiSCSIAccess=false
EnableFibreChannelAccess=true
```

See the PerformBackgroundLUNDuplicationResponse.txt file for example response data.

SEE ALSO

PerformLUNDuplication on p. 447

DeleteSnapLUN on p. 191

GetAllSnapLUNs on p. 233

GetAllSnapLUNHierarchies on p. 232

GetSnapLUNDetails on p. 297

ModifySnapLUN on p. 381

Ranges for Option Values on p. 57

NAME

PerformCancelTask

DESCRIPTION

Sends a cancellation request to stop the specified task. The Pillar Axiom storage system stops its processing when it receives the cancellation request, but does not roll back parts of the task that have already been completed.

SYNTAX

```
pdscli {request | submit} PerformCancelTask
   Task=GUID-or-FQN-of-task
```

OPTIONS

Task

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the task to cancel.

EXAMPLES

Run the following command to cancel a task.

```
pdscli request PerformCancelTask @input
```

where input.req contains the following name/value pairs:

```
Request=PerformCancelTask
Task=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the PerformCancelTaskResponse.txt file for example response data.

SEE ALSO

GetAllTasks on p. 235

Synchronous and Asynchronous Requests on p. 47 (some asynchronous requests cannot be cancelled)

Ranges for Option Values on p. 57

NAME **PerformClearLostData**

DESCRIPTION Indicates that permanent data loss likely occurred on the volume. Replacing the damaged hardware clears the flag and sets the volume back online to be recovered from a backup source. On NAS systems, an FSCK is run after the PerformClearLostData flag is cleared.

SYNTAX

```
pdscli {request | submit}
  PerformClearLostData
    AdministratorAction=GUID-or-FQN-of-action
    OwnerID=GUID-of-volume
```

OPTIONS

AdministratorAction

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of all administrator actions for which you want to display information.

OwnerID [Optional]

Identifies the owning volume. For NAS volumes, this is the filesystem ID. For SAN volumes, this is the LUN ID.

EXAMPLES Run the following command to display information about an unresolved administrator action.

```
pdscli request PerformClearLostData @input
```

where input.req contains the following name/value pairs:

```
Request= PerformClearLostData
AdministratorActionID=IDf1234567-89ab-cde0-
  0123-456789eeeee0
```

See the PerformClearLostDataResponse.txt file for example response data.

NAME

PerformCollectInstantaneousStatistics

DESCRIPTION

Collects real-time information about the Pillar Axiom system.

SYNTAX

```
pdscli {request | submit} PerformCollectInstantaneousStatistics
```

OPTIONS

None.

EXAMPLES

Run the following command to collect instantaneous data on the Pillar Axiom system.

```
pdscli request PerformCollectInstantaneousStatistics @input
```

where input.req contains the following name/value pairs:

```
PerformCollectInstantaneousStatistics
```

See the PerformCollectInstantaneousStatistics.txt file for example response data.

SEE ALSO

GetBackupRunStatistics on p. 241

GetFileServerStatistics on p. 254

GetFileSystemStatistics on p. 260

GetHistoricalFileSystemStatistics on p. 263

GetSlammerStatistics on p. 296

GetSystemStatusDetails on p. 303

PerformCollectSystemInformation on p. 407

Ranges for Option Values on p. 57

NAME **PerformCollectSystemInformation**

DESCRIPTION Collects information about the Pillar Axiom system and creates a tar file of trace logs. Pillar World Wide Customer Support Center may request that you perform this request to diagnose issues in your Pillar Axiom storage system.

SYNTAX

```
pdscli {request | submit} PerformCollectSystemInformation
  CollectTraceLogParameters.
    IncludeOnlyLatestLogs={true | false}
  CollectTraceLogParameters.
    IncludeInMemoryLogs={true | false}
  CollectTraceLogParameters.
    IncludeBackupLogs={true | false}
  CollectTraceLogParameters.
    IncludeAllHardwareComponents={true | false}
  CollectTraceLogParameters.
    IncludePilot={true | false}
  CollectTraceLogParameters.
    IncludeSANHostLogs={true | false}
  CollectTraceLogParameters.SlammerList.
    SlammerEntry.SlammerID=GUID-of-Slammer
  CollectTraceLogParameters.SlammerList.
    SlammerEntry.SlammerFQN=FQN-of-Slammer
  CollectTraceLogParameters.SlammerList.
    SlammerEntry.Name=string
  CollectTraceLogParameters.BrickList.
    BrickEntry.BrickID=GUID-of-Brick
  CollectTraceLogParameters.BrickList.
    BrickEntry.BrickFQN=FQN-of-Brick
  CollectTraceLogParameters.BrickList.
    BrickEntry.Name=string
  CollectTraceLogParameters.Days
    ZeroPlusIntDays=integer
  SendOff={true | false}
  SystemInformationType={DebugLogs |
    Statistics | RecentEvents
    |HardwareInventory | SoftwareInventory
    | NTP | ManagementConfiguration
    | SystemTopology | FileServers
    | VolumeGroups | SANInventory
    | Persistence | PDSFS}
```

OPTIONS **CollectTraceLogParameters.IncludeOnlyLatestLogs**

Identifies whether or not to collect only the most recent logs or all logs.

- Choose true to collect one Pilot log and one Slammer log (if the Slammer has generated any events).

- Choose false to collect all trace logs.

Important! You must enter DebugLogs for SystemInformationTypeChapter 4; otherwise, this option is invalid.

**CollectTraceLogParameters.
IncludeInMemoryLogs**

Identifies whether or not to write details from the utility's active memory to another trace log and include this log with the component-specific trace logs that you specify.

**CollectTraceLogParameters.
IncludeBackupLogs**

Identifies whether or not to write details from a backup copy of the utility's active memory to another trace log and include this log with the component-specific trace logs that you specify. The Pillar Axiom system periodically creates the backup copy, which may contain useful details to compare to the active-memory trace.

**CollectTraceLogParameters.
IncludeAllHardwareComponents**

Identifies whether or not to include trace logs from all hardware components. If you set this value to false, identify specific hardware components from which to gather trace logs. If you omit this value or set it to true, then all trace logs are collected.

CollectTraceLogParameters.IncludePilot

Identifies whether or not to include trace logs from the Pillar Axiom Pilot.

**CollectTraceLogParameters.IncludeSANHost
Logs**

Identifies whether to include SAN host logs from the Pillar Axiom Pilot.

**CollectTraceLogParameters.SlammerList.
SlammerEntry.SlammerID**

Enter the globally unique ID (GUID), which is a 38-character string that starts with “ID” and ends with 36 lower-case hex characters (a-f and 0-9) and hyphens.

Example GUID:

```
ID12345678-9abc-def0-1234-56789abcdef0
```

**CollectTraceLogParameters.SlammerList.
SlammerEntry.SlammerFQN**

Enter the fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

FQN format:

```
/Fully/Qualified/ObjectName
```

CollectTraceLogParameters.SlammerList.SlammerEntry.Name

Identifies the name that is assigned to a Slammer.

CollectTraceLogParameters.BrickList.BrickEntry.BrickID

Enter the globally unique ID (GUID), which is a 38-character string that starts with “ID” and ends with 36 lower-case hex characters (a-f and 0-9) and hyphens.

Example GUID:

```
ID12345678-9abc-def0-1234-56789abcdef0
```

CollectTraceLogParameters.BrickList.BrickEntry.BrickFQN=

Enter the fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name.

FQN format:

```
/Fully/Qualified/ObjectName
```

CollectTraceLogParameters.BrickList.BrickEntry.Name

Identifies the name that is assigned to a Brick.

CollectTraceLogParameters.Days (optional)

Specifies the number of days, from today, in which to collect the logs.

SendOff

Allows you to select whether you want the logs to be sent to the Call-Home server as soon as the collection operation completes.

SystemInformationType

Identifies the specific type of system information to collect. Choose from:

Table 24 System information type

Name	Description
“DebugLogs”	Software errors or hardware failures that may help diagnose system problems
“Statistics”	Performance statistics for filesystem backups, logical volumes, and NAS and SAN protocols
“RecentEvents”	High-level events that have occurred on the system. By default, this option returns either the last 30,000 events or events from the last two weeks, whichever is greater.

Table 24 System information type (continued)

Name	Description
“Hardware Inventory”	List of all hardware components, including serial numbers, in the Pillar Axiom storage system
“Software Inventory”	List of all software components in the Pillar Axiom storage system
NTP	Information about the Network Time Protocol (NTP) configured on the system
Management Configuration	Information about the global settings that are defined for the Pillar Axiom storage system
SystemTopology	List of all hardware components and internal typology on the Fibre Channel subset Note: This option does not return SCSI information, which you can access from the GUI.
FileServers	Information about the current File Server configuration, including defined routes, virtual interfaces, DNS settings, and NIS settings
VolumeGroups	Information about the current volume group configuration
SANInventory	List of the SAN components in the Pillar Axiom storage system, including every LUN, HBA port, SAN host, and SAN host LUN mapping.
Persistence	System configuration Information about system management and data resources as well as COD.
PDSFS	Information about the current filesystem configuration

EXAMPLES

Run the following command to create a trace log that contains details about your Pillar Axiom storage system.

```
pdscli request PerformCollectSystemInformation @input
```

where input.req contains the following name/value pairs:

```
PerformCollectSystemInformation
  CollectTraceLogParameters.
    IncludeOnlyLatestLogs=true
  CollectTraceLogParameters.
    IncludeInMemoryLogs=true
  CollectTraceLogParameters.
    IncludeBackupLogs=true
  CollectTraceLogParameters.
    IncludePilot=true
  CollectTraceLogParameters.
    IncludeSANHostLogs=true
  CollectTraceLogParameters.SlammerList.
    SlammerEntry.SlammerID=IDA0479DC2-38EF-9828-8902-E0FC03743270
  CollectTraceLogParameters.BrickList.
    BrickEntry.BrickID=IDA0479DC2-38EF-9828-
  CollectTraceLogParameters.Days
    ZeroPlusIntDays=1
  SendOff=true
  SystemInformationType=DebugLogs
```

See the PerformCollectSystemInformation.txt file for example response data.

SEE ALSO

GetAllCallHomeFileDetails on p. 205

GetBackupRunStatistics on p. 241

GetFileServerStatistics on p. 254

GetFileSystemStatistics on p. 260

GetHistoricalFileSystemStatistics on p. 263

GetSlammerStatistics on p. 296

GetSystemStatusDetails on p. 303

PerformCollectInstantaneousStatistics on p. 406

PerformDownloadCallHomeFile on p. 418

PerformDownloadSystemInformationFile on p. 420

PerformPeriodicCallHome on p. 460

PerformTestCallHome on p. 505

Ranges for Option Values on p. 57

NAME **PerformConfigureLUNMapping**

DESCRIPTION Maps a LUN to a specified SAN host.

SYNTAX

```
pdscli {request | submit} PerformConfigureLUNMapping
  MappedLUN=GUID-or-FQN-of-mapped-LUN
  HostMapping.SANHostName=Host-name
  HostMapping.MappingInformation.LUNNumber=
    integer-0-through-255
  HostMapping.MappingInformation.
    ExcludePortList.SlammerIOPort.
    ControlUnitNumber=
  HostMapping.MappingInformation.
    ExcludePortList.SlammerIOPort.
    FibreChannelPortName=PORT-name
```

OPTIONS

MappedLUN

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID of the LUN to map.

HostMapping.SANHostName

Identifies the name assigned to a specific SAN host.

HostMapping.MappingInformation.LUNNUmber

Identifies the unique number that is assigned to a LUN and can be accessed by all SAN hosts if the LUN is not mapped. If the LUN is mapped, the LUN number must be unique to the mapped SAN host.

Hostmapping.MappingInformation.ExcludePortList.SlammerIOPort.ControlUnitNumber

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the control unit.

Hostmapping.MappingInformation.ExcludePortList.SlammerIOPort.FibreChannelPortName

Identifies a fibre channel port name.

Note: This value is case-sensitive; you must use capitalized letters for PORT, FC0, or FS0.

EXAMPLES

Run the following command to map EngLUN to the SANTest_02 host.

```
pdscli request PerformConfigureLUNMapping @input
```

where input.req contains the following name/value pairs:

```
Request=PerformConfigureLUNMapping
MappedLUN=IDA0479DC2-38EF-9828-8902-
     E0FC03743270
HostMapping.SANHostName=SANTest_02
HostMapping.MappingInformation.LUNNumber=0
HostMapping.MappingInformation.
     ExcludePortList.SlammerIOPort.
     ControlUnitNumber=0
HostMapping.MappingInformation.
     ExcludePortList.SlammerIOPort.
     FibreChannelPortName=PORT1
```

See the PerformConfigureLUNMappingResponse.txt file for example response data.

SEE ALSO

CreateLUN on p. 135

DeleteLUN on p. 182

GetAllLUNs on p. 221

GetHostMappingConfigDetails on p. 265

GetLUNDetails on p. 269

GetLUNMappingConfigDetails on p. 270

ModifyLUN on p. 351

PerformRebuildLUNRedundancy on p. 475

Ranges for Option Values on p. 57

NAME **PerformCopybackBrickLUNs**

DESCRIPTION Initiates a copyback to a Brick LUN.

SYNTAX

```
pdscli {request | submit} PerformRebuildBrickLUNs
  BrickWWN=world-wide-name
  TargetDriveNumber={0 | integer}
  SourceDriveNumber={0 | integer}
  LUNID=GUID-or-FQN-of-LUN
```

OPTIONS

BrickWWN

Identifies the unique identifier, or world wide name (WWN), of the specified Brick.

TargetDriveNumber

Identifies the target disk drive that is installed on the Brick.

SourceDriveNumber

Identifies the source disk drive that is installed on the Brick.

LUNID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick LUN to rebuild.

EXAMPLES Run the following command to rebuild a LUN on a Brick.

```
pdscli request PerformCopybackBrickLUNs @input
```

where input.req contains the following name/value pairs:

```
Request=PerformCopybackBrickLUNs
LUNID=/EngLUN
TargetDriveNumber=
SourceDriveNumber=
BrickWWN=21:00:00:e0:8:11:b5:4a
```

See the PerformCopybackBrickLUNsResponse.txt file for example response data.

SEE ALSO

GetAllBricks (p. 204)

GetBrickDetails (p. 242)

PerformRebuildBrickLUNs (p. 472)

NAME **PerformDisableSMIStartup**

DESCRIPTION Disables Storage Management Initiative - Specification (SMI-S), which has been manually enabled on a Pilot. SMI-S is a messaging-based interface designed to support the specific requirements of managing devices in a SAN. By default, all 512 MB-based Pilots have SMI-S disabled on startup. Once enabled, SMI-S will remain on the Pilot through power cycles until it is manually disabled.

SYNTAX

```
pdscli {request | submit} PerformDisableSMIStartup
```

OPTIONS None.

EXAMPLES Run the following command to manually disable SMI-S.

```
pdscli request PerformDisableSMIStartup @input
```

where input.req contains the following name/value pairs:

```
Request=PerformDisableSMIStartup
```

See the PerformDisableSMIStartupResponse.txt file for example response data.

SEE ALSO PerformEnableSMIStartup on p. 421

NAME

PerformDownloadCallHomeFile

DESCRIPTION

Downloads a tar file with Call Home information, including the system configuration, status, and events, from the Pillar Axiom storage system to your administrative workstation. Transmit the file as requested by World Wide Customer Support Center.

You can also use this request to download an event-generated Call Home file, if Call Home is not configured or something is preventing Call Home from transferring the file automatically.

SYNTAX

```
pdscli {request | submit}
  PerformDownloadCallHomeFile
    FileNameOnly={true | false}
```

OPTIONS

FileNameOnly

Identifies whether to include the name of the tar file in the command response. You must set this option to false.

EXAMPLES

Run the following command to download system information.

```
pdscli request PerformDownloadCallHomeFile @input
```

where input.req contains the following name/value pairs:

```
PerformDownloadCallHomeFile
  FileNameOnly=false
```

See the PerformDownloadCallHomeFileResponse.txt file for example response data.

SEE ALSO

GetAllCallHomeFileDetails on p. 205

GetSystemInformationFileDetails on p. 302

GetSystemStatusDetails on p. 303

PerformPeriodicCallHome on p. 460

PerformTestCallHome on p. 505

NAME **PerformDownloadQuotaFile**

DESCRIPTION Downloads the requested quota report file to the user’s client. Obtain the list of available quota reports using the GetAllQuotaFiles request.

SYNTAX

```
pdscli {request | submit} PerformDownloadQuotaFile
   ID=GUID-or-FQN-of-quota
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem for which you want to generate a quota report.

EXAMPLES Run the following command to display information about a quota that is defined for the engineering organization’s filesystem, EngAtl.

```
pdscli request PerformDownloadQuotaFile @input
```

where input.req contains the following name/value pairs:

```
Request=PerformDownloadQuotaFile
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the PerformDownloadQuotaFileResponse.txt file for example response data.

SEE ALSO CreateQuota on p. 154

DeleteQuota on p. 186

GatherQuotaForFileSystem on p. 197

GetAllQuotas on p. 224

GetAllQuotaFiles on p. 225

GetQuotaDetails on p. 282

ModifyQuota on p. 375

NAME

PerformDownloadSystemInformationFile

DESCRIPTION

Downloads a tar file with statistics from the Pillar Axiom storage system to your administrative workstation. Transmit the file as requested by World Wide Customer Support Center.

SYNTAX

```
pdscli {request | submit}
  PerformDownloadSystemInformationFile
    SystemInformationFile=name-of-system-information-file
```

OPTIONS

SystemInformationFile [optional]

Identifies the name of the file that you want to download.

EXAMPLES

Run the following command to download system information.

```
pdscli request PerformDownloadSystemInformationFile @input
```

where input.req contains the following name of the request:

```
PerformDownloadSystemInformationFile
```

See the PerformDownloadSystemInformationFileResponse.txt file for example response data.

SEE ALSO

GetBackupRunStatistics on p. 241

GetFileServerStatistics on p. 254

GetFileSystemStatistics on p. 260

GetHistoricalFileSystemStatistics on p. 263

GetSlammerStatistics on p. 296

GetSystemStatusDetails on p. 303

PerformCollectInstantaneousStatistics on p. 406

PerformCollectSystemInformation on p. 407

Ranges for Option Values on p. 57

NAME **PerformEnableSMIStartup**

DESCRIPTION Enables Storage Management Initiative - Specification (SMI-S) on a Pilot. SMI-S is a messaging-based interface designed to support the specific requirements of managing devices in a SAN.

Important! By default, all 512 MB-based Pilots have SMI-S, VDS, and VSS disabled on startup. Because of limited memory, you should not enable SMI on these Pilots. For anything other than testing purposes, contact your Pillar Account Representative for a memory upgrade to your Pilot.

Note: You do not need to run this command on 2 GB-based Pilots.

Once the PerformEnableSMIStartup command has been executed, contact Pillar World Wide Customer Service to complete the process.

When the system comes back online, the smProvider service should be enabled. Once enabled, the service will remain on through power cycles until you manually disable it with PerformDisableSMIStartup.

SYNTAX

```
pdscli {request | submit} PerformEnableSMIStartup
```

OPTIONS None.

EXAMPLES Run the following command to manually enable SMI-S.

```
pdscli request PerformEnableSMIStartup @input
```

where input.req contains the following name/value pairs:

```
Request=PerformEnableSMIStartup
```

See the PerformEnableSMIStartupResponse.txt file for example response data.

SEE ALSO PerformDisableSMIStartup on p. 417

NAME **PerformFailbackResources**

DESCRIPTION Initiates a failback of a Slammer CU that is in FailedOver status. This option can also be used to recover from a Slammer CU Failover on a NAS Slammer, if auto failback is disabled in the server configuration.

SYNTAX

```
pdscli {request | submit} PerformFailbackResources
    Slammer=GUID-or-FQN-of-Slammer
    ControlUnitNumber={0 | 1}
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer.

ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

EXAMPLES Run the following command to fail back CU1 on Slammer1 if CU1 is currently in FailedOver status and automatic failback is not enabled.

```
pdscli request PerformFailbackResources @input
```

where input.req contains the following name/value pairs:

```
Request=PerformFailbackResources
Slammer=/Company-XZY-data/Slammer1
ControlUnitNumber=1
```

See the PerformFailbackResourcesResponse.
txt file for example response data.

NAME **PerformFileSystemCheck**

DESCRIPTION Checks the consistency of the specified filesystem.

SYNTAX

```
pdscli {request | submit} PerformFileSystemCheck
   FileSystem=GUID-or-FQN-of-filesystem
   FileSystemSnapshot=snap-FS-name
   StopOnFirstCorruption={true | false}
```

OPTIONS

FileSystem

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem to check.

FileSystemSnapshot

Identifies the name that the Pillar Axiom system assigns to a Snap FS.

StopOnFirstCorruption

Halts the filesystem check if a "filesystem corrupted" response is detected. This option can be used to end the fsck to begin a recovery from a snapshot, file system copy, NDMP backup, or data replication repository.

Note: You can check the GetAllAdministratorActionsResponse.txt file for "filesystem corrupted" responses.

EXAMPLES Run the following command to check the consistency of the OpsLA filesystem after you received an AdministratorAction.

```
pdscli request PerformFileSystemCheck @input
```

where input.req contains the following name/value pairs:

```
Request=PerformFileSystemCheck
FileSystem=/Company-XYZ-FS1/OpsLA
Name=EngAt1-snap
StopOnFirstCorruption=true
```

See the PerformFileSystemCheckResponse.txt file for example response data.

See the GetAllAdministratorActionsResponse.txt file for “filesystem corrupted” responses.

SEE ALSO

PerformRestoreFileSystemFromSnapshot on p. 485

Ranges for Option Values on p. 57

NAME **PerformFileSystemDuplication**

DESCRIPTION Creates a filesystem duplicate, which is defined as:

> A block-level, duplicate copy (image copy) of a filesystem. Use this data redundancy feature (called Copy Filesystem in the Pillar Axiom Storage Services Manager) to create a bit-for-bit secondary duplicate of a source filesystem for immediate use as a read-write copy. Note that the Quality of Service (QoS) may differ from the original.
>
> A duplicate copy requires greater system and storage resources than a Snap FS. To create an archival copy, create an inactive clone.

Use this request, for example, to replicate a data set for testing or to set up a new environment. User data, NFS exports, and CIFS shares are duplicated. You can change the capacity and performance settings of the duplicate filesystem.

Note: This request does not automatically start a background copy. To start a background copy, see the PerformBackgroundFileSystemDuplication request.

SYNTAX

```
pdscli {request | submit} PerformFileSystemDuplication
   SourceFileSystem=GUID-or-FQN-of-filesystem
   FileServer=GUID-or-FQN-of-File-Server
   VolumeGroup=GUID-or-FQN-of-volume-group
   Name=string
   Profile={Normal | HighThroughput}
   MaximumCapacity=integer
   RelativePriority={Archive | Low |Medium | High | Premium}
   FileSizeBias={Small | Medium |Large | None}
   AccessBias={None | Mixed |Random | Sequential}
   IOBias={Mixed | Write | Read | None}
   StorageClass={MultiLevelCellSolidStateDrives |
      SingleLevelCellSolidStateDrives | FibreChannelHardDiskDrives |
      SataHardDiskDrives}
   Redundancy={Standard | Double}
   AccessMode=ReadWrite
   CloneFSCapacity=interger
   EnforceVolumeRetention={true | false}
   BackgroundCopyPriority=
      {MaximizeSpeed | MinimizeImpact | SystemChooses}
```

OPTIONS

SourceFileSystem

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the source filesystem to duplicate.

FileServer

Enter the GUID or FQN of a File Server from which the new filesystem gets protocol-access options.

VolumeGroup

Enter the GUID or FQN of the volume group with which the new filesystem is associated.

Name

Identifies the name that is assigned to a duplicated filesystem. The new filesystem's name must be unique within the File Server and volume group to which the duplicated filesystem is assigned.

Profile [optional]

Identifies a performance profile. Choose from:

- Normal, which defines specific performance and redundancy settings for each logical volume. This value is highly recommended and should be used in almost all cases.
- OracleASM, which uses a wide stripe RAID geometry consisting of 1 MB strips to improve the overall performance of logical volumes in an Oracle ASM environment.
- HighThroughput, which stripes the filesystem or LUN across all RAID arrays. SATA Bricks have two RAID groups per Brick, and FC Bricks have one RAID group per Brick. You do not need to set double redundancy to get full array striping.

Important! Use of the HighThroughput profile is for special purposes only. Do not use this profile for ordinary filesystems or LUNs. Please contact the Pillar Customer Services Advanced Solutions Group before using this profile option, which should only be used for special testing where the system configuration is cleared after the test.

MaximumCapacity

Identifies the maximum capacity, in MB, to which a given object can grow.

RelativePriority

The Relative Priority determines how much of the system resources are devoted to that volume, including the allocation of Slammer CPU cycles and the allocation of specific portions of the disk platters. The higher the Priority, the greater the allocation of CPU time and the faster the media access time.

The Relative Priority option prioritizes the layout of data that is stored in the Pillar Axiom system based on one of the following settings:

- Premium priority is the highest possible performance. This priority uses the outer 20% of the drive platters.

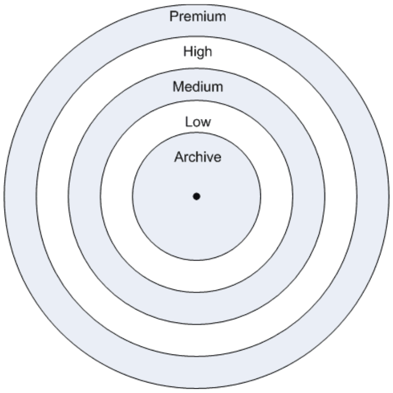
Note: I/O is typically faster when data resides on the outer edge of the drive platters.

- High priority logical volumes are allocated space in the outer 20%-40% of the drive platters.

Tip: If sufficient Bricks are present, performance of high priority volumes is enhanced by placing those volumes on a greater number of Bricks.

- Medium priority logical volumes are allocated space in the outer 40% to 60% of the drive platters.
- Low priority logical volumes are allocated space in the band that is 60% to 80% from the outer diameter of drive platters.
- Archive (lowest-priority) logical volumes are allocated space in the inner areas of the drive platters.

Figure 5 Example of HDD priority bands



If needed, the Pillar Axiom system allocates space in a higher or lower priority band (within the same Storage Class) than the one that you choose.

FileSizeBias

Identifies the typical size of files that are stored in the specified logical volume. Choose from:

- Small if files are smaller than 20 kilobytes (KB).
- Medium if files are larger than 20KB and smaller than 4 megabytes (MB).
- Large if files are larger than 4MB.

AccessBias

Identifies the typical data access method. Choose from:

Table 25 Data access methods

Access bias	I/O bias	Read ahead in the Slammer	RAID configuration in the Brick
Sequential	Read	Aggressive	RAID 5 Reads large extents of the data into memory.
	Write	Conservative	RAID 5 • Allocates capacity in large, full-stripe (640 K) extents. • Writes data to physical storage more quickly and in full-stripe extents.
Mixed and random	Read	None	RAID 5 • Reads small extents into memory. • Allocates capacity in small chunks. • Writes data to physical storage more slowly to combine (or eliminate) write operations.
Random	Write	None	RAID 10

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

IOBias

Identifies the typical read-write ratio. Choose from:

- Read if users or applications read data more often than they write to the data source.
- Write if users or applications write data more often than they read it.

Important! If you choose Random as the access method and Write as the I/O Bias, the system creates the volume with a Distributed RAID geometry. This geometry provides enhanced write performance but uses twice the capacity.

- Mixed if the read-write ratio varies.

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

The system stores all writes of user data and system metadata in mirrored copies of the journal. Although the filesystem or LUN is journaled, for some of the profiles, write caching is disabled because it tends to have very low hit rates.

One copy is maintained in non-volatile memory on one control unit (CU) of a Slammer. The mirror copy is maintained in one of:

- Battery-backed memory of the partner CU on the Slammer (preferred location). Writes to this copy are equivalent to write-back cache.
- An area reserved on disk for the logical volume if the partner CU is unavailable for the write. Writes to this copy are equivalent to write-through cache.

Writes from the journal to permanent disk storage are equivalent to write-through cache. The system flushes user data and the corresponding metadata as a unit to disk.

StorageClass

Identifies the category of physical storage on which the logical volume resides:

- Fibre Channel drives
- SATA drives
- SATA single-level cell solid state drives
- SATA multi-level cell solid state drives

Redundancy

Identifies how many mirror copies of the original data are stored online.

Important! Pillar highly recommends that you consult with a Pillar customer support professional for assistance with sizing your system and creating your volumes (filesystems and LUNs).

Redundancy options include:

- Standard, to store original data only. Data striping over multiple RAID groups maintains full redundancy, even without mirror copies.
- Double, to store original data and one mirror copy, with data striping over multiple RAID groups.

Note: Double Redundancy can only provide true redundancy if your system has enough Bricks to allocate the filesystem or LUN such that no two mirror copies share a RAID group.

SATA Bricks have two RAID groups per Brick, FC Bricks have one RAID group per brick.

If the storage pool is becoming depleted, or a large filesystem or LUN is created, it might be necessary to place the filesystem or LUN on more RAID group fragments.

Depending on the system's ability to allocate sufficient contiguous storage blocks for the size of the logical volume, use the following number of RAID groups to configure your volumes for best performance:

Table 26 Optimum number of RAID groups for best performance

Priority	SATA standard Redundancy	SATA double redundancy	FS standard redundancy	FS double redundancy
Archive	4	8	2	4
Low	4	8	2	4
Medium	6	12	3	6
High	8	16	4	8
Premium	8	16	4	8

For *performance testing purposes only*, create a filesystem or LUN using standard redundancy and the High Throughput Profile. This is NOT recommended for most applications. Reset your system after you have created a High Throughput LUN or filesystem before you configure normal filesystems or LUNs for applications.

AccessMode

Identifies whether the filesystem is made available to users in read-write mode. Read-write mode means that users can read data from the filesystem and write new data to it.

Note: Read-only AccessMode is no longer supported.

CloneFSCapacity

Specifies the maximum amount of repository space to be reserved for Clone FSs from the total system capacity.

⚠ CAUTION

We strongly recommend that you allocate sufficient repository capacity to minimize the chances of running out of this space (which could lead to data inconsistency or loss). To set sufficient capacity, use a value equal to the source volume capacity times the number of replicas times the maximum rate of change. For example, for a 100 GB volume that is projected to have 20 active replicas at a time and no more than a 20% rate of change, use a value of 400 GB for the clone repository.

Maximum capacity for Clone FSs is 2147483647.

EnforceVolumeRetention

Forces the retention policy settings, which protect files on a filesystem.

BackgroundCopyPriority

Identifies the strategy the system should use to control the impact on performance when background tasks need to copy or move data from one location in the storage pool to another:

- MinimizeImpact. Restricts the amount of work performed on a loaded system. This option is intended to have a minimal impact on client I/O throughput at the expense of longer copy times.
- SystemChooses. Balances the background copy with the incoming client I/O. This option is the default.
- MaximizeSpeed. Prioritizes the background copy at the expense of client I/O throughput.

Note: When the system is idle or lightly loaded, the above background task maximizes the amount of work done regardless of the option selected.

The following types of operations are affected by the strategy you select:

- Copy
- Restore
- Quality of Service (QoS) Priority changes
- Storage Class changes

Data transfer operations invoked by the Pillar Axiom MaxRep Replication for NAS utility are not affected by the BackgroundCopyPriority setting.

EXAMPLES

Run the following command to duplicate the EngAtl filesystem, plus its NFS exports and CIFS shares.

```
pdscli request PerformFileSystemDuplication @input
```

where input.req contains the following name/value pairs:

```
Request=PerformFileSystemDuplication
SourceFileSystem=/Company-XYZ-FS1/EngAtl
FileServer=/Company-XYZ-FS1
VolumeGroup=/Eng
Name=EngAtlNew
Profile=Normal
MaximumCapacity=500
RelativePriority=Archive
FileSizeBias=Small
AccessBias=Sequential
IOBias=Write
StorageClass=FibreChannelHardDiskDrives
```

```
Redundancy=Standard
AccessMode=ReadWrite
BackgroundCopyPriority=MaximizeSpeed
```

See the PerformFileSystemDuplicationResponse.txt file for example response data.

SEE ALSO

PerformBackgroundFileSystemDuplication on p. 400

CreateFileSystemSnapshotSchedule on p. 124

DeleteFileSystemSnapshot on p. 177

DeleteFileSystemSnapshotSchedule on p. 179

GetAllFileSystemSnapshots on p. 216

GetAllFileSystemSnapshotSchedules on p. 218

GetFileSystemSnapshotDetails on p. 256

GetFileSystemSnapshotScheduleDetails on p. 258

ModifyFileSystemSnapshot on p. 340

PerformRestoreFileSystemFromSnapshot on p. 485

PerformRebuildFileSystemRedundancy on p. 474

Ranges for Option Values on p. 57

NAME **PerformIdentifyBrick**

DESCRIPTION To determine that maintenance activity is being performed on the correct Brick, identifies the Brick by blinking the front panel LEDs. A reverse action can identify all Bricks *except* for the one that needs servicing.

SYNTAX

```
pdscli {request | submit} PerformIdentifyBrick
  Brick=GUID-or-FQN-of-Brick
  IdentifyCommand={Beacon | Inverse | Cancel}
```

OPTIONS

Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick to be serviced.

IdentifyCommand

Identifies how to blink LEDs to identify a hardware component, or a field replaceable unit (FRU) in a component for Guided Maintenance. Choose from:

- Beacon, which tells the Pillar Axiom storage system to blink the LEDs on the Brick bezel.
- Inverse, which tells the Pillar Axiom storage system to blink all other Bricks instead of the one that needs to be serviced. This option blinks Bricks only; it does not blink Slammers.
- Cancel, which tells the system to stop blinking the LEDs. Choose this option after you have located the hardware component or FRU.

EXAMPLES Run the following command to flash the LEDs on all Bricks except the one that needs to be serviced.

```
pdscli request PerformIdentifyBrick @input
```

where input.req contains the following name/value pairs:

```
Request=PerformIdentifyBrick
Brick=/Company-XZY-data/Brick3
IdentifyCommand=Inverse
```

See the PerformIdentifyBrickResponse.txt file for example response data.

SEE ALSO

PerformAcceptAddBrick on p. 391

PerformResumeBrick on p. 486

Ranges for Option Values on p. 57

Service Guide

NAME **PerformIdentifyBrickFRU**

DESCRIPTION Identifies a Brick with a field replaceable unit (FRU) that needs to be serviced. Pillar Axiom storage systems blink LEDs to identify Brick FRUs that need to be serviced. The LEDs for the drives are on the front of the Brick, behind the bezel. The LEDs for the SATA spare are in the back of the Brick, as are those for the power supply, the ES module, and the RAID controller. For the FC Brick, the spare is floating and the LED is on the front.

SYNTAX

```
pdscli {request | submit} PerformIdentifyBrickFRU
  Brick=GUID-or-FQN-of-Brick
  FRUName={PowerSupplyFanModule | ESModule |
    DiskDrive | SpareDiskDrive | RAIDController |
    Chassis}
  FRUNumber=integer
  IdentifyCommand={Beacon | Inverse | Cancel}
```

OPTIONS

Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick on which a FRU is to be serviced.

FRUName

Identifies a field replaceable unit (FRU) on a Brick. Choose from:

- PowerSupplyFanModule, which specifies either of the two power supply/fan modules that are installed on a Brick.
- ESModule, which specifies the enclosure services (ES) module that is installed on a Brick.
- DiskDrive, which specifies any of the six disk drives that are installed on each Brick control unit (CU).
- SpareDiskDrive, which specifies the spare disk drive that is shared by the two Brick CUs.
- RAIDController, which specifies either of the two RAID controllers that are installed on a Brick.
- Chassis, which specifies an entire Brick CU.

FRUNumber

Identifies a field replaceable unit (FRU) on a hardware component that is installed on the Pillar Axiom system.

IdentifyCommand

Identifies how to blink LEDs to identify a hardware component, or a field replaceable unit (FRU) in a component for Guided Maintenance. Choose from:

- Beacon, which tells the Pillar Axiom storage system to blink the LEDs on the Brick bezel.
- Inverse, which tells the Pillar Axiom storage system to blink all other Bricks instead of the one that needs to be serviced. This option blinks Bricks only; it does not blink Slammers.
- Cancel, which tells the system to stop blinking the LEDs. Choose this option after you have located the hardware component or FRU.

EXAMPLES

Run the following command to flash the LEDs on all Bricks except the one that contains a FRU that needs to be serviced.

```
pdscli request PerformIdentifyBrickFRU @input
```

where input.req contains the following name/value pairs:

```
Request=PerformIdentifyBrickFRU
Brick=/Company-XZY-data/Brick3
FRUName=DiskDrive
FRUNumber=42
IdentifyCommand=Inverse
```

See the PerformIdentifyBrickFRUResponse.txt file for example response data.

SEE ALSO

PerformPrepareReplaceBrickFRU on p. 461

PerformResumeBrickFRU on p. 487

Ranges for Option Values on p. 57

Service Guide

NAME **PerformIdentifyPilot**

DESCRIPTION Identifies a Pilot control unit (CU) that needs to be serviced. Pillar Axiom storage systems blink LEDs to identify Pilot control units that need to be serviced. The beacon for the Pilot blinks the red hard drive LED behind the bezel; there is no blinking of anything on the bezel itself.

The LED that is blinked is the red disk drive activity LED. It is very difficult to see the difference between the blinking of an identify Pilot and normal Pilot-to-Pilot CU synchronization activities. Because of this difficulty, contact a Pillar Support Center representative, who is skilled at detecting the difference.

SYNTAX

```
pdscli {request | submit} PerformIdentifyPilot
   PilotUnitName={pilot1 | pilot2}
   IdentifyCommand={Beacon | Inverse | Cancel}
```

OPTIONS **PilotUnitName**

Identifies a specific control unit (CU) in a Pilot. Each Pilot contains two CUs.

IdentifyCommand

Identifies how to blink LEDs to identify a hardware component, or a field replaceable unit (FRU) in a component for Guided Maintenance. Choose from:

- Beacon, which tells the Pillar Axiom storage system to blink the LEDs on the Brick bezel.
- Inverse, which tells the Pillar Axiom storage system to blink all other Bricks instead of the one that needs to be serviced. This option blinks Bricks only; it does not blink Slammers.
- Cancel, which tells the system to stop blinking the LEDs. Choose this option after you have located the hardware component or FRU.

EXAMPLES Run the following command to flash the LEDs on a Pilot control unit that needs to be serviced.

```
pdscli request PerformIdentifyPilot @input
```

where input.req contains the following name/value pairs:

```
Request=PerformIdentifyPilot
PilotUnitName=pilot1
IdentifyCommand=Beacon
```

See the PerformIdentifyPilotResponse.txt file for example response data.

SEE ALSO ModifyManagementConfig on p. 360

Ranges for Option Values on p. 57

Service Guide

NAME

PerformIdentifySlammer

DESCRIPTION

Identifies a Slammer that needs to be serviced. Pillar Axiom storage systems blink LEDs to identify Slammers that need to be serviced. The LEDs for the Slammer are on the bezel, on the rear, upper left corner of the FCIM/PIM.

SYNTAX

```
pdscli {request | submit} PerformIdentifySlammer
   Slammer=GUID-or-FQN-of-Slammer
   ControlUnitNumber={0 | 1}
   IdentifyCommand={Beacon | Inverse | Cancel}
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer to be serviced.

ControlUnitNumber [optional]

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs. The CUs are identified by 0 (the top CU) and 1 (the bottom CU).

IdentifyCommand

Identifies how to blink LEDs to identify a hardware component, or a field replaceable unit (FRU) in a component for Guided Maintenance. Choose from:

- Beacon, which tells the Pillar Axiom storage system to blink the LEDs on the Brick bezel.
- Inverse, which tells the Pillar Axiom storage system to blink all other Bricks instead of the one that needs to be serviced. This option blinks Bricks only; it does not blink Slammers.
- Cancel, which tells the system to stop blinking the LEDs. Choose this option after you have located the hardware component or FRU.

EXAMPLES

Run the following command to flash the LEDs on all Slammers except the one that needs to be serviced.

```
pdscli request PerformIdentifySlammer @input
```

where input.req contains the following name/value pairs:

```
request=PerformIdentifySlammer
Slammer=/Company-XZY-data/Slammer1
ControlUnitNumber=1
IdentifyCommand=Inverse
```

See the PerformIdentifySlammerResponse.txt file for example response data.

SEE ALSO

PerformAcceptAddSlammer on p. 392

PerformResumeSlammer on p. 489

Ranges for Option Values on p. 57

Service Guide

NAME **PerformIdentifySlammerFRU**

DESCRIPTION Identifies a Slammer with a field replaceable unit (FRU) that needs to be serviced. Pillar Axiom storage systems blink LEDs to identify Slammer FRUs that need to be serviced.

SYNTAX

```
pdscli {request | submit} PerformIdentifySlammerFRU
   Slammer=GUID-or-FQN-of-Slammer
   ControlUnitNumber={0 | 1}
      FRUName={PowerSupply | FanModule |
      Battery | Memory | Motherboard |
      NASNetworkInterfaceModule |
      SANNetworkInterfaceModule |
      PrivateInterconnectModule |
      SCSIController | Chassis}
   FRUNumber=integer
   IdentifyCommand={Beacon | Inverse | Cancel}
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer on which a FRU is to be serviced.

ControlUnitNumber [optional]

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

FRUName

Identifies a field replaceable unit (FRU) on a Slammer. Choose from:

- PowerSupply, which specifies either of the two power supplies that are installed on each Slammer control unit (CU). Each Slammer contains four power supplies.
- FanModule, which specifies either of the two fan assemblies that are installed on each Slammer CU. Each fan assembly contains two fans (which are not individual FRUs). The fan module LED is in the front of the Slammer, behind the bezel.

- Battery, which specifies either of the two batteries that are installed on each Slammer CU. The battery is located in the front of the Slammer, behind the bezel. There is only one battery per CU.
- Memory, which specifies memory modules that reside on the motherboard that is installed on each Slammer CU. There are no memory LEDs.
- Motherboard, which specifies the motherboard that is installed on each Slammer CU.
- NASNetworkInterfaceModule, which specifies either of the two network interface modules that are installed on a NAS Slammer. Both Slammer control units (CUs) must contain NAS or SAN network interface modules. The NAS module is located in the rear of the Slammer.
- SANNetworkInterfaceModule, which specifies either of the two network interface modules that are installed on a SAN Slammer. The SAN module is located in the rear of the Slammer.
- PrivateInterconnectModule (PIM), which specifies either of the two private interconnect modules that are installed on a Slammer. The PIM is located in the rear of the Slammer.
- SCSIController, which specifies the PCI card that resides in the network interface module that is installed on each Slammer CU. The SCSIController is in the rear, inside the SAN or NAS module.
- Chassis, which specifies an entire Slammer CU.

FRUNumber [optional]

Identifies a field replaceable unit (FRU) on a hardware component that is installed on the Pillar Axiom system.

IdentifyCommand

Identifies how to blink LEDs to identify a hardware component, or a field replaceable unit (FRU) in a component for Guided Maintenance. Choose from:

- Beacon, which tells the Pillar Axiom storage system to blink the LEDs on the Brick bezel.
- Inverse, which tells the Pillar Axiom storage system to blink all other Bricks instead of the one that needs to be serviced. This option blinks Bricks only; it does not blink Slammers.
- Cancel, which tells the system to stop blinking the LEDs. Choose this option after you have located the hardware component or FRU.

EXAMPLES

Run the following command to flash the LEDs on all Slammers except the one that contains a FRU that needs to be serviced.

```
pdscli request PerformIdentifySlammerFRU @input
```

where input.req contains the following name/value pairs:

```
Request=PerformIdentifySlammerFRU
Slammer=/Company-XZY-data/Slammer1
```

```
ControlUnitNumber=1
FRUName=NASNetworkInterfaceModule
FRUNumber=42
IdentifyCommand=Inverse
```

See the PerformIdentifySlammerFRUResponse.txt file for example response data.

SEE ALSO

PerformPrepareReplaceSlammerFRU on p. 463

PerformResumeSlammerFRU on p. 490

Ranges for Option Values on p. 57

Service Guide

NAME **PerformIdentifySlammerIOPort**

DESCRIPTION Identifies the physical network port that is associated with a File Server's virtual network interface.

SYNTAX

```
pdscli {request | submit}
   PerformIdentifySlammerIOPort
      Slammer=GUID-or-FQN-of-Slammer
      EthernetPortName={PORT0 | PORT1}
      IOPortNumber={0 | 1}
      IdentifyCommand={Beacon | Inverse | Cancel}
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Slammer that contains the physical network port that is associated with a particular virtual network interface.

ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

EthernetPortName

Identifies the physical network port in a network interface module to use for the identification operation.

IdentifyCommand

Identifies how to blink LEDs to identify a hardware component, or a field replaceable unit (FRU) in a component for Guided Maintenance. Choose from:

- Beacon, which tells the Pillar Axiom storage system to blink an LED on the hardware component that needs to be serviced.
- Inverse, which tells the system to blink LEDs on all components except the hardware component that needs to be serviced.
- Cancel, which tells the system to stop blinking the LEDs. Choose this option after you have located the hardware component or FRU.

EXAMPLES Run the following command to identify a virtual network interface on the Slammer that is named Slammer1.

```
pdscli request PerformIdentifySlammerIOPort
   @input
```

where input.req contains the following name/value pairs:

```
Request=PerformIdentifySlammerIOPort
Slammer=/Company-XZY-data/Slammer1
ControlUnitNumber=0
EthernetPortName= PORT0
IdentifyCommand=Beacon
```

See the PerformIdentifySlammerIOPortResponse.txt file for example response data.

SEE ALSO

GetAllSlammers on p. 231

GetAllVirtualInterfaces on p. 239

Ranges for Option Values on p. 57

Service Guide

NAME **PerformJoinDomain**

DESCRIPTION Joins the specified File Server to the CIFS domain with which that File Server is associated.

SYNTAX

```
pdscli {request | submit} PerformJoinDomain
   FileServer=GUID-or-FQN-of-File-Server
   DomainAdminUsername=string
   DomainAdminPassword=string
```

OPTIONS **FileServer**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the File Server to join to a CIFS domain.

DomainAdminUsername

Identifies a user name for the File Server to access the Domain Controller when anonymous access is not permitted. This account should be:

- Similar to the Guest account, with the same type of low-security privileges.
- Different from the Administrator account for the Domain Controller.

DomainAdminPassword

Identifies the user password for the Pillar Axiom File Server to access the CIFS Domain Controller if anonymous access is not permitted.

EXAMPLES Run the following command to have the Company-XYZ-FS1 File Server join the CIFS domain that is associated with the File Server.

```
pdscli request PerformJoinDomain @input
```

where input.req contains the following name/value pairs:

```
Request=PerformJoinDomain
FileServer=/Company-XYZ-FS1
DomainAdminUsername=admin
DomainAdminPassword=password
```

See the PerformJoinDomainResponse.txt file for example response data.

SEE ALSO CreateFileServer on p. 100

ModifyFileServer on p. 321

Ranges for Option Values on p. 57

NAME **PerformLUNDuplication**

DESCRIPTION Creates a LUN duplicate, which is defined as:

> A block-level, image copy of a source LUN used for immediate read-write access. A LUN duplicate (called Copy LUN in the Pillar Axiom Storage Services Manager) creates a reliable, bit-for-bit secondary copy of the entire allocated storage space of the source LUN.
>
> This feature requires greater system and storage resources than the Snap LUN feature. To create an archival copy, create an inactive clone.

Use this request, for example, to replicate a data set for testing or to set up a new environment. You can change the capacity and performance settings of the duplicate LUN.

Note: The PerformLUNDuplication request automatically copies and duplicates the LUN in one operation.

SYNTAX

```
pdscli {request | submit} PerformLUNDuplication
   SourceLUN=GUID-or-FQN-of-source-LUN
   Name=string
   VolumeGroup=GUID-or-FQN-of-volume-group
   LUNNumber=integer-0-through-255
   Mapped={true | false}
   EnableiSCSIAccess={true | false}
   EnableFibreChannelAccess={true | false}
   Profile={Normal | HighThroughput}
   Capacity=integer
   Max=integer
   RelativePriority={High | Medium | Low |
      Archive}
   AccessBias={Sequential | Random | Mixed}
   IOBias={Read | Write | Mixed}
   StorageClass={MultiLevelCellSolidStateDrives |
   SingleLevelCellSolidStateDrives |
      FibreChannelHardDiskDrives | SataHardDiskDrives}
   Redundancy={Standard | Double |}
   SnapLUNStorage.MaximumCapacity=integer
   BackgroundCopyPriority=
   {MaximizeSpeed | MinimizeImpact | SystemChooses}
```

OPTIONS

SourceLUN

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the source LUN to duplicate.

Name

Identifies the name that is assigned to a LUN for administrative purposes. LUN names must be unique across the Pillar Axiom system, not just within its associated volume group.

Name is a required parameter only if you want to change the FQN LUN's user-visible name.

VolumeGroup

Enter the GUID or FQN of the volume group with which the new LUN is associated.

LUNNumber [optional]

Identifies the unique number that is assigned to a LUN and can be accessed by all SAN hosts if the LUN is not mapped. If the LUN is mapped, the LUN number must be unique to the mapped SAN host.

Note: If this LUN is mapped to a specified host, do not include the LUNNumber option.

Mapped [optional]

Identifies whether all hosts or only selected hosts may access this LUN.

- Mapped allows only designated hosts to access this LUN.
- Unmapped allows all hosts to access this LUN using the LUN Number.

Note: If you choose Mapped, do not assign a LUNNumber.

EnableiSCSIAccess

Identifies the protocol specification used when creating a SAN LUN. Options include iSCSI, Fibre Channel (FC), or both.

EnableFibreChannelAccess

Identifies the protocol specification used when creating a SAN LUN. Options include iSCSI, Fibre Channel (FC), or both.

Profile [optional]

Identifies a performance profile. Choose from:

- Normal, which defines specific performance and redundancy settings for each logical volume. This value is highly recommended and should be used in almost all cases.
- OracleASM, which uses a wide stripe RAID geometry consisting of 1 MB strips to improve the overall performance of logical volumes in an Oracle ASM environment.
- HighThroughput, which stripes the filesystem or LUN across all RAID arrays. SATA Bricks have two RAID groups per Brick, and FC Bricks have one RAID group per Brick. You do not need to set double redundancy to get full array striping.

Important! Use of the HighThroughput profile is for special purposes only. Do not use this profile for ordinary filesystems or LUNs. Please contact the Pillar Customer Services Advanced Solutions Group before using this profile option, which should only be used for special testing where the system configuration is cleared after the test.

Capacity

Identifies the capacity that is assigned to this LUN.

Note: Once submitted, you cannot modify this value.

RelativePriority

The Relative Priority determines how much of the system resources are devoted to that volume, including the allocation of Slammer CPU cycles and the allocation of specific portions of the disk platters. The higher the Priority, the greater the allocation of CPU time and the faster the media access time.

The Relative Priority option prioritizes the layout of data that is stored in the Pillar Axiom system based on one of the following settings:

- Premium priority is the highest possible performance. This priority uses the outer 20% of the drive platters.

Note: I/O is typically faster when data resides on the outer edge of the drive platters.

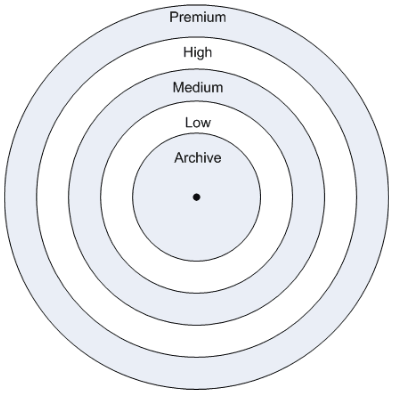
- High priority logical volumes are allocated space in the outer 20%-40% of the drive platters.

Tip: If sufficient Bricks are present, performance of high priority volumes is enhanced by placing those volumes on a greater number of Bricks.

- Medium priority logical volumes are allocated space in the outer 40% to 60% of the drive platters.
- Low priority logical volumes are allocated space in the band that is 60% to 80% from the outer diameter of drive platters.

- Archive (lowest-priority) logical volumes are allocated space in the inner areas of the drive platters.

Figure 6 Example of HDD priority bands



If needed, the Pillar Axiom system allocates space in a higher or lower priority band (within the same Storage Class) than the one that you choose.

AccessBias

Identifies the typical data access method. Choose from:

Table 27 Data access methods

Access bias	I/O bias	Read ahead in the Slammer	RAID configuration in the Brick
Sequential	Read	Aggressive	RAID 5 Reads large extents of the data into memory.
	Write	Conservative	RAID 5 • Allocates capacity in large, full-stripe (640 K) extents. • Writes data to physical storage more quickly and in full-stripe extents.
Mixed and random	Read	None	RAID 5 • Reads small extents into memory. • Allocates capacity in small chunks. • Writes data to physical storage more slowly to combine (or eliminate) write operations.
Random	Write	None	RAID 10

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

IOBias

Identifies the typical read-write ratio. Choose from:

- Read if users or applications read data more often than they write to the data source.
- Write if users or applications write data more often than they read it.

Important! If you choose Random as the access method and Write as the I/O Bias, the system creates the volume with a Distributed RAID geometry. This geometry provides enhanced write performance but uses twice the capacity.

- Mixed if the read-write ratio varies.

Note: This is an optimization bias, not a requirement that all data or data operations conform to the value.

The system stores all writes of user data and system metadata in mirrored copies of the journal. Although the filesystem or LUN is journaled, for some of the profiles, write caching is disabled because it tends to have very low hit rates.

One copy is maintained in non-volatile memory on one control unit (CU) of a Slammer. The mirror copy is maintained in one of:

- Battery-backed memory of the partner CU on the Slammer (preferred location). Writes to this copy are equivalent to write-back cache.
- An area reserved on disk for the logical volume if the partner CU is unavailable for the write. Writes to this copy are equivalent to write-through cache.

Writes from the journal to permanent disk storage are equivalent to write-through cache. The system flushes user data and the corresponding metadata as a unit to disk.

StorageClass

Identifies the category of physical storage on which the logical volume resides:

- Fibre Channel drives
- SATA drives
- SATA single-level cell solid state drives
- SATA multi-level cell solid state drives

Redundancy

Identifies how many mirror copies of the original data are stored online.

Important! Pillar highly recommends that you consult with a Pillar customer support professional for assistance with sizing your system and creating your volumes (filesystems and LUNs).

Redundancy options include:

- Standard, to store original data only. Data striping over multiple RAID groups maintains full redundancy, even without mirror copies.
- Double, to store original data and one mirror copy, with data striping over multiple RAID groups.

Note: Double Redundancy can only provide true redundancy if your system has enough Bricks to allocate the filesystem or LUN such that no two mirror copies share a RAID group.

SATA Bricks have two RAID groups per Brick, FC Bricks have one RAID group per brick.

If the storage pool is becoming depleted, or a large filesystem or LUN is created, it might be necessary to place the filesystem or LUN on more RAID group fragments.

Depending on the system's ability to allocate sufficient contiguous storage blocks for the size of the logical volume, use the following number of RAID groups to configure your volumes for best performance:

Table 28 Optimum number of RAID groups for best performance

Priority	SATA standard Redundancy	SATA double redundancy	FS standard redundancy	FS double redundancy
Archive	4	8	2	4
Low	4	8	2	4
Medium	6	12	3	6
High	8	16	4	8
Premium	8	16	4	8

For *performance testing purposes only*, create a filesystem or LUN using standard redundancy and the High Throughput Profile. This is NOT recommended for most applications. Reset your system after you have created a High Throughput LUN or filesystem before you configure normal filesystems or LUNs for applications.

SnapLUNStorage.MaximumCapacity

Identifies the maximum capacity, in MB, to which a given object can grow.

BackgroundCopyPriority

Identifies the strategy the system should use to control the impact on performance when background tasks need to copy or move data from one location in the storage pool to another:

- MinimizeImpact. Restricts the amount of work performed on a loaded system. This option is intended to have a minimal impact on client I/O throughput at the expense of longer copy times.
- SystemChooses. Balances the background copy with the incoming client I/O. This option is the default.
- MaximizeSpeed. Prioritizes the background copy at the expense of client I/O throughput.

Note: When the system is idle or lightly loaded, the above background task maximizes the amount of work done regardless of the option selected.

The following types of operations are affected by the strategy you select:

- Copy
- Restore
- Quality of Service (QoS) Priority changes
- Storage Class changes

Data transfer operations invoked by the Pillar Axiom MaxRep Replication for NAS utility are not affected by the BackgroundCopyPriority setting.

EXAMPLES

Run the following command to create EngLUNnew as a duplicate of EngLUN.

```
pdscli request PerformLUNDuplication @input
```

where input.req contains the following name/value pairs:

```
Request=PerformLUNDuplication
SourceLUN=/EngLUN
Name=EngLUNnew
VolumeGroup=/Eng
Profile=Normal
Capacity=1
RelativePriority=Archive
AccessBias=Sequential
IOBias=Write
StorageClass=FibreChannelHardDiskDrives
Redundancy=Standard
SnapLUNStorage.MaximumCapacity=0
EnableiSCSIAccess=false
EnableFibreChannelAccess=true
BackgroundCopyPriority=MaximizeSpeed
```

See the PerformLUNDuplicationResponse.txt file for example response data.

SEE ALSO

PerformBackgroundLUNDuplication on p. 402

DeleteSnapLUN on p. 191

GetAllSnapLUNs on p. 233

GetAllSnapLUNHierarchies on p. 232

GetSnapLUNDetails on p. 297

ModifySnapLUN on p. 381

Ranges for Option Values on p. 57

NAME **PerformNISFileDownload**

DESCRIPTION Downloads a tar file that contains a set of previously uploaded NIS-alternative files from the Pillar Axiom storage system to your administrative workstation.

SYNTAX

```
pdscli {request | submit} PerformNISFileDownload
   FileServer=GUID-or-FQN-of-File-Server
   FileNameOnly={true | false}
```

OPTIONS

FileServer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the File Server with which the NIS-alternative files are associated.

FileNameOnly [optional]

Identifies whether to include the name of the tar file in the command response.

- Enable this option so that you know the name of the tar file.
- Disable this option if you do not need the name of the tar file. Choose this option if you plan to use the CLI to download the file.

EXAMPLES Run the following command to download the files that are used to authenticate NFS users.

```
pdscli request PerformNISFileDownload @input
```

where input.req contains the following name/value pairs:

```
Request=PerformNISFileDownload
FileServer=/Company-XYZ-FS1
FileNameOnly=false
```

See the PerformNISFileDownloadResponse.txt file for example response data.

SEE ALSO

GetNISUploadedFilesDetails on p. 277

PerformNISFileUpload on p. 456

Ranges for Option Values on p. 57

NAME **PerformNISFileUpload**

DESCRIPTION Encodes and loads the specified /etc/passwd, /etc/group, and /etc/netgroup files from a network machine onto the Pillar Axiom Pilot. The File Server consults the uploaded files to authenticate NFS users if you do not use Network Information Service (NIS) for NFS authentication.

SYNTAX

```
pdscli {request | submit} PerformNISFileUpload
   FileServer=GUID-or-FQN-of-File-Server
   PasswordFile.UserFilename=filename-string
   PasswordFile.InternalFilename=string
   PasswordFile.FileContents="&@filename;"
   GroupFile.UserFilename=filename-string
   GroupFile.InternalFilename=string
   GroupFile.FileContents="&@filename;"
   NetgroupFile.UserFilename=filename-string
   NetgroupFile.InternalFilename=string
   NetgroupFile.FileContents="&@filename;"
   HostsFile.UserFilename=filename-string
   HostsFile.InternalFilename=string
   HostsFile.FileContents="&@filename;"
```

OPTIONS

FileServer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the File Server with which the NIS-alternative files are associated.

PasswordFile.UserFilename

Identifies the name of the /etc/passwd file to Base64 encode and upload to the Pillar Axiom storage system.

PasswordFile.InternalFilename

Identifies a name for the uploaded NIS-alternative file. If you have multiple File Servers and you plan to use NIS-alternative files for each, this identifier uniquely identifies a specific File Server's files.

PasswordFile.FileContents

Identifies that the Pillar Axiom system should Base64 encode the contents of the specified file and upload the encoded file. The `&`, `@` and `;` characters that surround the file name tell the CLI to encode and upload the file.

GroupFile.UserFilename

Identifies the name of the /etc/group file to Base64 encode and upload to the Pillar Axiom storage system.

GroupFile.InternalFilename

Identifies a name for the uploaded NIS-alternative file. If you have multiple File Servers and you plan to use NIS-alternative files for each, this identifier uniquely identifies a specific File Server's files.

GroupFile.FileContents

Identifies that the Pillar Axiom system should Base64 encode the contents of the specified file and upload the encoded file. The `&`, `@` and `;` characters that surround the file name tell the CLI to encode and upload the file.

NNetgroupFile.UserFilename

Identifies the name of the /etc/netgroup file to Base64 encode and upload to the Pillar Axiom storage system.

NetgroupFile.InternalFilename

Identifies a name for the uploaded NIS-alternative file. If you have multiple File Servers and you plan to use NIS-alternative files for each, this identifier uniquely identifies a specific File Server's files.

NetgroupFile.FileContents

Identifies that the Pillar Axiom system should Base64 encode the contents of the specified file and upload the encoded file. The `&`, `@` and `;` characters that surround the file name tell the CLI to encode and upload the file.

HostsFile.UserFilename

Identifies the name of the /etc/netgroup file to Base64 encode and upload to the Pillar Axiom storage system.

HostsFile.InternalFilename

Identifies a name for the uploaded NIS-alternative file. If you have multiple File Servers and you plan to use NIS-alternative files for each, this identifier uniquely identifies a specific File Server's files.

HostsFile.FileContents

Identifies that the Pillar Axiom system should Base64 encode the contents of the specified file and upload the encoded file. The `&`, `@` and `;` characters that surround the file name tell the CLI to encode and upload the file.

EXAMPLES

Run the following command to upload NIS-alternative files to the Pilot in your system.

```
pdscli request PerformNISFileUpload @input
```

where input.req contains the following name/value pairs:

```
Request=PerformNISFileUpload
FileServer=/Company-XYZ-FS1
```

```
PasswordFile.UserFilename=passwd
PasswordFile.InternalFilename=FS1-passwd
PasswordFile.FileContents="&@passwd;"
GroupFile.UserFilename=group
GroupFile.InternalFilename=FS1-group
GroupFile.FileContents="&@group;"
NetgroupFile.UserFilename=netgroup
NetgroupFile.InternalFilename=FS1-netgroup
NetgroupFile.FileContents="&@netgroup;"
HostsFile.UserFilename=hosts
HostsFile.InternalFilename=FS1-hosts
HostsFile.FileContents="&@nhosts;"
```

See the PerformNISFileUploadResponse.txt file for example response data.

SEE ALSO

GetNISUploadedFilesDetails on p. 277

PerformNISFileDownload on p. 455

Ranges for Option Values on p. 57

NAME **PerformNLMLockRecovery**

DESCRIPTION Performs a Network Lock Manager Recovery. The NLM works in cooperation with the Network File System (NFS) to provide a System V style of advisory file and record locking over the network. The network lock manager and the network status monitor are network-service daemons. Both daemons are essential to the kernel's ability to provide fundamental network services.

SYNTAX

```
pdscli {request | submit} PerformNLMLockRecovery
  FileServerFQN=FQN-of-File-Server
```

OPTIONS FileServerFQN

EXAMPLES Run the following command to display status information for your system.

```
pdscli request PerformNLMLockRecovery
  FileServerFQN=/Company-XYZ-FS1
```

See the PerformNLMLockRecovery.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **PerformPeriodicCallHome**

DESCRIPTION The Call-Home feature is defined as:

A Pillar Axiom feature that, when enabled, allows the system to notify the Pillar World Wide Customer Support Center of critical issues specific to Pillar Axiom system. No customer data is transmitted. The Call-Home feature transfers files over the Internet using one of the following user-selected methods:

- SCP: Uses secure copy (SCP) with 1024-bit encryption and secure keys.
- HTTPS: Sends files either directly to the Pillar World Wide Customer Support Center or through a proxy server for security purposes. Can also be used when the Pillar Axiom system does not have direct access to the Internet.

SYNTAX

```
pdscli {request | submit}
  PerformPeriodicCallHome
```

OPTIONS None.

EXAMPLES Run the following command to create a Call-Home file to send to the Pillar World Wide Customer Support Center for analysis.

```
pdscli request PerformPeriodicCallHome
```

See the PerformPeriodicCallHomeResponse.txt file for example response data.

SEE ALSO PerformTestCallHome on p. 505

Contacts at Pillar Data Systems on p. 29

NAME

PerformPrepareReplaceBrickFRU

DESCRIPTION

Prepares the Pillar Axiom storage system for:

- Removal of a field replaceable unit (FRU) in a Brick hardware component.
- Addition of a new FRU to replace the old one.

SYNTAX

```
pdscli {request | submit}
  PerformPrepareReplaceBrickFRU
    Brick=GUID-or-FQN-of-Brick
    FRUName={PowerSupplyFanModule | ESModule
      | DiskDrive | SpareDiskDrive |
      RAIDController | Chassis}
    FRUNumber=integer
```

OPTIONS

Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Brick on which a FRU will be replaced when the system is ready.

FRUName

Identifies a field replaceable unit (FRU) on a Brick. Choose from:

- PowerSupplyFanModule, which specifies either of the two power supply/fan modules that are installed on a Brick.
- ESModule, which specifies the enclosure services (ES) module that is installed on a Brick.
- DiskDrive, which specifies any of the six disk drives that are installed on each Brick control unit (CU).
- SpareDiskDrive, which specifies the spare disk drive that is shared by the two Brick CUs.
- RAIDController, which specifies either of the two RAID controllers that are installed on a Brick.
- Chassis, which specifies an entire Brick CU.

FRUNumber [optional]

Identifies a field replaceable unit (FRU) on a hardware component that is installed on the Pillar Axiom system.

EXAMPLES

Run the following command to prepare your system for replacement of one of the array disk drives on a Brick.

```
pdscli request PerformPrepareReplaceBrickFRU @input
```

where input.req contains the following name/value pairs:

```
Request=PerformPrepareReplaceBrickFRU
Brick=/Company-XYZ-data/Brick3
FRUName=DiskDrive
FRUNumber=42
```

See the PerformPrepareReplaceBrickFRUResponse.txt file for example response data.

SEE ALSO

PerformIdentifyBrickFRU on p. 435

PerformResumeBrickFRU on p. 487

PerformAcceptForeignDiskDrive on p. 395

Ranges for Option Values on p. 57

Service Guide

NAME **PerformPrepareReplaceSlammerFRU**

DESCRIPTION Prepares the Pillar Axiom storage system for:

- Removal of a field replaceable unit (FRU) in a Slammer hardware component.
- Addition of a new FRU to replace the old one.

SYNTAX

```
pdscli {request | submit}
  PerformPrepareReplaceSlammerFRU
    Slammer=GUID-or-FQN-of-Slammer
    ControlUnitNumber={0 | 1}
    FRUName={PowerSupply | FanModule |
      Battery |
      Memory | Motherboard |
      NASNetworkInterfaceModule |
      SANNetworkInterfaceModule |
      PrivateInterconnectModule |
      SCSIController | Chassis}
    FRUNumber=integer
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Slammer on which a FRU will be replaced when the system is ready.

ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

FRUName

Identifies a field replaceable unit (FRU) on a Slammer. Choose from:

- PowerSupply, which specifies either of the two power supplies that are installed on each Slammer control unit (CU). Each Slammer contains four power supplies.
- FanModule, which specifies either of the two fan assemblies that are installed on each Slammer CU. Each fan assembly contains two fans (which are not individual FRUs).
- Battery, which specifies either of the two batteries that are installed on each Slammer CU.

- Memory, which specifies memory modules that reside on the motherboard that is installed on each Slammer CU.
- Motherboard, which specifies the motherboard that is installed on each Slammer CU.
- NASNetworkInterfaceModule, which specifies either of the two network interface modules that are installed on a NAS Slammer. Both Slammer control units (CUs) must contain NAS or SAN network interface modules.
- SANNetworkInterfaceModule, which specifies either of the two network interface modules that are installed on a SAN Slammer.
- PrivateInterconnectModule, which specifies either of the two private interconnect modules that are installed on a Slammer.
- SCSIController, which specifies the PCI card that resides in the network interface module that is installed on each Slammer CU.
- Chassis, which specifies an entire Slammer CU.

FRUNumber [optional]

Identifies a field replaceable unit (FRU) on a hardware component that is installed on the Pillar Axiom system.

EXAMPLES

Run the following command to prepare your system for replacement of a network interface module on a Slammer.

```
pdscli request PerformPrepareReplaceSlammerFRU
   @input
```

where input.req contains the following name/value pairs:

```
Request=PerformPrepareReplaceSlammerFRU
Slammer=/Company-XZY-data/Slammer1
ControlUnitNumber=1
FRUName=NASNetworkInterfaceModule
FRUNumber=42
```

See the PerformPrepareReplaceSlammerFRU Response.txt file for example response data.

SEE ALSO

PerformIdentifySlammerFRU on p. 440

PerformResumeSlammerFRU on p. 490

Ranges for Option Values on p. 57

Service Guide

NAME **PerformPrepareSlammerMemoryUpgrade**

DESCRIPTION Prepares the Pillar Axiom storage system for a memory replacement or upgrade on a Slammer storage controller.

SYNTAX

```
pdscli {request | submit}
   PerformPrepareSlammerMemoryUpgrade
      Slammer=GUID-or-FQN-of-Slammer
      MemorySize={3 | 6 | 12}
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Slammer on which the memory will be upgraded when the system is ready.

Memory

Identifies the amount of Slammer memory in gigabytes. Choose from:

- 3
- 6
- 12

Both control units (CUs) in a Slammer must be configured with the same amount of memory.

EXAMPLES Run the following command to prepare your system for a memory upgrade on a Slammer.

```
pdscli request PerformPrepareSlammerMemoryUpgrade
   @input
```

where input.req contains the following name/value pairs:

```
Request=PerformPrepareSlammerMemoryUpgrade
Slammer=/Company-XZY-data/Slammer1
MemorySize=12
```

See the PerformPrepareSlammerMemoryUpgrade Response.txt file for example response data.

SEE ALSO PerformIdentifySlammerFRU on p. 440

PerformResumeSlammerFRU on p. 490

Ranges for Option Values on p. 57

Service Guide

NAME **PerformProtectedFileValidation**

DESCRIPTION

This scan checks all protected files on the filesystem and validates their data integrity and occurs on demand.

See the event log for scan results.

SYNTAX

```
pdscli {request | submit}
  PerformProtectedFileValidation
    ID=GUID-or-FQN-of-filesystem
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem to put online and return it to service.

EXAMPLES

Run the following command to return the engineering organization's filesystem to service.

```
pdscli request PerformProtectedFileValidation @input
```

where input.req contains the following name/value pairs:

```
Request=PerformProtectedFileValidation
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the PerformProtectedFileValidationResponse.txt file for example response data.

SEE ALSO

Ranges for Option Values on p. 57

GetAllAdministratorActions on p. 201

GetAdministratorActionDetails on p. 199

PerformFileSystemCheck on p. 423

ReenableAllRAIDControllers on p. 521

ReenableAllSlammers on p. 522

ReenableCUBoot on p. 523

NAME

PerformPutFileSystemOffline

DESCRIPTION

Puts the specified filesystem offline so that users cannot access the filesystem data. You may have to perform this request before you check filesystem consistency.

SYNTAX

```
pdscli {request | submit}
  PerformPutFileSystemOffline
    ID=GUID-or-FQN-of-filesystem
```

OPTIONS

ID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem to put online and return it to service.

EXAMPLES

Run the following command to return the engineering organization's filesystem to service.

```
pdscli request PerformPutFileSystemOffline @input
```

where input.req contains the following name/value pairs:

```
Request=PerformPutFileSystemOffline
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the PerformPutFileSystemOfflineResponse.txt file for example response data.

SEE ALSO

Ranges for Option Values (p. 57)

GetAllAdministratorActions (p. 201)

GetAdministratorActionDetails (p. 199)

PerformFileSystemCheck (p. 423)

PerformPutFileSystemOnline (p. 470)

ReenableAllRAIDControllers (p. 521)

NAME **PerformPutFileSystemOnline**

DESCRIPTION Puts the specified filesystem online so that users can access the filesystem data again. You may have to perform this request after you respond to an action-required notification to check filesystem consistency.

SYNTAX

```
pdscli {request | submit}
  PerformPutFileSystemOnline
    ID=GUID-or-FQN-of-filesystem
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem to put online and return it to service.

EXAMPLES Run the following command to return the engineering organization's filesystem to service.

```
pdscli request PerformPutFileSystemOnline @input
```

where input.req contains the following name/value pairs:

```
Request=PerformPutFileSystemOnline
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the PerformPutFileSystemOnlineResponse.txt file for example response data.

SEE ALSO Ranges for Option Values (p. 57)

GetAllAdministratorActions (p. 201)

GetAdministratorActionDetails (p. 199)

PerformFileSystemCheck (p. 423)

ReenableAllRAIDControllers (p. 521)

ReenableAllSlammers (p. 522)

ReenableCUBoot (p. 523)

NAME **PerformRebuildBrickLUNs**

DESCRIPTION Rebuilds a Brick LUN.

SYNTAX

```
pdscli {request | submit} PerformRebuildBrickLUNs
   BrickWWN=world-wide-name
   TargetDriveNumber={0 | integer}
   LUNID=GUID-or-FQN-of-LUN
```

OPTIONS

BrickWWN

Identifies the unique identifier, or world wide name (WWN), of the specified Brick.

TargetDriveNumber

Identifies the target disk drive that is installed on the Brick.

LUNID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick LUN to rebuild.

EXAMPLES Run the following command to rebuild a LUN on a Brick.

```
pdscli request PerformRebuildBrickLUNs @input
```

where input.req contains the following name/value pairs:

```
Request=PerformRebuildBrickLUNs
LUNID=/EngLUN
TargetDriveNumber=0
BrickWWN=21:00:00:e0:8:11:b5:4a
```

See the PerformRebuildBrickLUNsResponse.txt file for example response data.

SEE ALSO GetAllBricks (p. 204)

GetBrickDetails (p. 242)

PerformCopybackBrickLUNs (p. 415)

NAME **PerformRebuildCloneFSRedundancy**

DESCRIPTION Rebuilds a doubly-redundant Clone FS that has degraded because a number of disk drives have failed, compromising the redundancy.

SYNTAX

```
pdscli {request | submit}
  PerformRebuildCloneFSRedundancy
    ID=GUID-or-FQN-of-CloneFS
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Clone FS to duplicate.

EXAMPLES Run the following command to display information for the FSClone_01 Clone FS.

```
pdscli request PerformRebuildCloneFSRedundancy @input
```

where input.req contains the following name/value pairs:

```
Request=PerformRebuildCloneFSRedundancy
Source=/FSClone_01
```

See the PerformRebuildCloneFSRedundancyResponse.txt file for example response data.

SEE ALSO CreateCloneFS (p. 97)

DeleteCloneFS (p. 172)

GetAllCloneFSs (p. 208)

GetCloneFSDetails (p. 245)

ModifyCloneFS (p. 317)

NAME

PerformRebuildFileSystemRedundancy

DESCRIPTION

Rebuilds a copy of a filesystem in the event that you lose one or more copies of data.

SYNTAX

```
pdscli {request | submit}
   PerformRebuildFileSystemRedundancy
         ID=GUID-or-FQN-of-filesystem
```

OPTIONS

ID

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem to rebuild.

EXAMPLES

Run the following command to rebuild the EngAtl filesystem.

```
pdscli request PerformRebuildFileSystemRedundancy
   @input
```

where input.req contains the following name/value pairs:

```
Request=PerformRebuildFileSystemRedundancy
FileSystem=/Company-XYZ-FS1/EngAtl
```

See the PerformRebuildFileSystemRedundancyResponse.txt file for example response data.

SEE ALSO

Ranges for Option Values on p. 57

NAME **PerformRebuildLUNRedundancy**

DESCRIPTION Rebuilds a LUN.

SYNTAX

```
pdscli {request | submit}
  PerformRebuildLUNRedundancy
    ID=GUID-or-FQN-of-LUN
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the LUN to duplicate.

EXAMPLES Run the following command to rebuild EngLUN.

```
pdscli request PerformRebuildLUNRedundancy
  @input
```

where input.req contains the following name/value pairs:

```
Request=PerformRebuildLUNRedundancy
LUN=/EngLUN
```

See the PerformRebuildLUNRedundancyResponse.txt file for example response data.

SEE ALSO

CreateLUN on p. 135

DeleteLUN on p. 182

GetAllLUNs on p. 221

GetHostMappingConfigDetails on p. 265

GetLUNDetails on p. 269

GetLUNMappingConfigDetails on p. 270

ModifySnapLUN on p. 381

PerformConfigureLUNMapping on p. 412

Ranges for Option Values on p. 57

NAME **PerformRebuildSnapLUNRedundancy**

DESCRIPTION Rebuilds a doubly-redundant Snap LUN that has degraded because a number of disk drives have failed, compromising the redundancy.

SYNTAX

```
pdscli {request | submit}
  PerformRebuildSnapLUNRedundancy
    ID=GUID-or-FQN-of-Snap-LUN
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Snap LUN to duplicate.

EXAMPLES Run the following command to display information for the SLUN_01 Snap LUN

```
pdscli request PerformRebuildSnapLUNRedundancy @input
```

where input.req contains the following name/value pairs:

```
Request=PerformRebuildSnapLUNRedundancy
Source=/SLUN_01
```

See the PerformRebuildSnapLUNRedundancyResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **PerformRejectForeignDiskDrive**

DESCRIPTION Rejects a new disk drive that was installed on a Brick. After a disk drive has been rejected, it can no longer be used in a Pillar Axiom system without first contacting the Pillar Worldwide Customer Support Center.

SYNTAX

```
pdscli {request | submit}
   PerformRejectForeignDiskDrive
      Brick=GUID-or-FQN-of-Brick
      DiskDriveNumber=integer
```

OPTIONS

Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick that contains the disk drive to reject.

DiskDriveNumber

Identifies a disk drive on a Brick that is installed on the Pillar Axiom system.

EXAMPLES Run the following command to reject a new disk drive when you receive a Foreign Drive Administrator Action.

```
pdscli request PerformRejectForeignDiskDrive @input
```

where input.req contains the following name/value pairs:

```
Request=PerformRejectForeignDiskDrive
Brick=/Company-XYZ-data/Brick3
DiskDriveNumber=42
```

See the PerformRejectForeignDiskDriveResponse.txt file for example response data.

SEE ALSO PerformIdentifyBrickFRU on p. 435

PerformPrepareReplaceBrickFRU on p. 461

PerformResumeBrickFRU on p. 487

Ranges for Option Values on p. 57

Service Guide

NAME **PerformRemoveBrickFromConfiguration**

DESCRIPTION Removes a Brick from your configuration when the Brick status is Missing.

Important! Before running this request, make sure to

- Remove all user data and persistence resources from the Brick.
- Power down the Brick.
- Wait until GetAllBricks or the GUI shows that the Brick status is Missing.

SYNTAX

```
pdscli {request | submit} PerformRemoveBrickFromConfiguration
Brick=GUID-or-FQN-of-Brick
```

OPTIONS **Brick**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Brick to remove.

EXAMPLES

```
Run the following command to remove Brick3 from the configuration.
pdscli request PerformRemoveBrickFromConfiguration@input
```

where input.req contains the following name/value pairs:

```
Request=PerformRemoveBrickFromConfiguration
Brick=/Company-XYZ-data/Brick3
```

See the PerformRemoveBrickFromConfigurationResponse.txt file for example response data.

SEE ALSO PerformPrepareReplaceBrickFRU on p. 461

Ranges for Option Values on p. 57

NAME **PerformRemoveSlammerFromConfiguration**

DESCRIPTION Removes a Slammer from your configuration when the Slammer status is Missing. You cannot run this request to replace a Slammer that has any data resources configured on the Slammer.

Before running this request, remove all volumes (filesystems or LUNs) from the Slammer, then power down the Slammer. Optionally, you can re-cable the private interface module (PIM) and the private management interface (PMI). Finally, wait for the Slammer status to show Missing.

Note: This request cannot be used on a Pillar Axiom 300 Slammer or single Slammer systems.

SYNTAX

```
pdscli {request | submit} PerformRemoveSlammerFromConfiguration
Slammer=GUID-or-FQN-of-Slammer
```

OPTIONS **Slammer**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer to remove.

EXAMPLES

```
Run the following command to remove Slammer1 from the configuration.
pdscli request PerformRemoveSlammerFromConfiguration@input
```

where input.req contains the following name/value pairs:

```
Request=PerformRemoveSlammerFromConfiguration
Slammer=/Company-XYZ-data/Slammer1
```

```
See the PerformRemoveSlammerFromConfigurationResponse.txt file for
example response data.
```

SEE ALSO PerformPrepareReplaceSlammerFRU on p. 463

Ranges for Option Values on p. 57

NAME **PerformResetSystem**

DESCRIPTION Deletes **all** user data and configuration details on the Pillar Axiom storage system. To perform this request, you must contact the Pillar World Wide Customer Support Center to obtain an encrypted file with the system serial number (SSN).

Note: You must provide the request to reset the system in writing.

SYNTAX

```
pdscli {request | submit} PerformResetSystem
   SystemSerialNumberFileName=Name-of-file
   SystemSerialNumberFileContents=
      "&@filename;"
   EncryptedSystemSerialNumber=string
```

OPTIONS

SystemSerialNumberFileName

Identifies the name of file that you receive from Technical Support, which contains the encrypted system serial number.

SystemSerialNumberFileContents [optional]

Identifies that the Pillar Axiom system should Base64 encode the contents of the specified file and upload the encoded file. The `&`, `@` and `;` characters that surround the file name tell the CLI to encode and upload the file.

EncryptedSystemSerialNumber [optional]

Identifies the encrypted system serial number (SSN) that is assigned to the Pillar Axiom storage system.

EXAMPLES Run the following command to delete all of the test data and configuration details on your system before moving the system into production.

```
pdscli request PerformResetSystem @input
```

where input.req contains the following name/value pairs:

```
Request=PerformResetSystem
SystemSerialNumberFileName=Company-XYZ-ssn
SystemSerialNumberFileContents="&@XYZ-ssn;"
EncryptedSystemSerialNumber=A000001BCD
```

See the PerformResetSystemResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **PerformResetTapeRobotDevice**

DESCRIPTION Resets the specified tape device or robot.

SYNTAX

```
pdscli {request | submit}
   PerformResetTapeRobotDevice
      TapeRobotFQN=GUID-or-FQN-of-tape-device
```

OPTIONS **TapeRobotFQN**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the tape device or robot that is to be reset.

EXAMPLES Run the following command to reset a tape device that is attached to your system.

```
pdscli request PerformResetTapeRobotDevice @input
```

where input.req contains the following name/value pairs:

```
Request=PerformResetTapeRobotDevice
TapeRobotFQN=IDf1234567-89ab-cde0-0123-
   456789eeeee0
```

See the PerformResetTapeRobotDeviceResponse.txt file for example response data.

SEE ALSO GetAllTapeRobots on p. 234

Ranges for Option Values on p. 57

NAME **PerformRestart**

DESCRIPTION Restarts the Pillar Axiom storage system after repairs or updates.

Note: For an immediate shutdown and restart, it is unnecessary to send the PerformShutdown request before the PerformRestart request; however, we recommend issuing a PerformShutdown followed by a GetCoreState to make sure the Pillar Axiom system has shutdown before issuing PerformRestart.

Tip: For a restart after an extended period of time, remember to reinstall the batteries before you restart the system.

SYNTAX

```
pdscli {request | submit} PerformRestart
```

OPTIONS None.

EXAMPLES Run the following command to restart you system after you increased Slammer memory from 3 GB to 12 GB.

```
pdscli request PerformRestart
```

See the PerformRestartResponse.txt file for example response data.

SEE ALSO PerformShutdown on p. 495

Ranges for Option Values on p. 57

NAME **PerformRestoreFileSystemFromSnapshot**

DESCRIPTION Restores the configuration and content (data files) of a filesystem from the specified Snap FS.

SYNTAX

```
pdscli {request | submit}
  PerformRestoreFileSystemFromSnapshot
    ID=GUID-or-FQN-of-Snap-FS
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Snap FS from which to restore the filesystem data.

EXAMPLES Run the following command to restore the EngAtl filesystem from a Snap FS.

```
pdscli request PerformRestoreFileSystemFromSnapshot
  @input
```

where input.req contains the following name/value pairs:

```
Request=PerformRestoreFileSystemFromSnapshot
ID=IDA0479DC2-38EF-9828-8902-E0FC03743270
```

See the PerformRestoreFileSystemFromSnapshot Response.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **PerformResumeBrick**

DESCRIPTION Resumes operations in the Pillar Axiom storage system after the:

- Removal of a Brick chassis.
- Addition of a new Brick to replace the old one.

SYNTAX

```
pdscli {request | submit} PerformResumeBrick
   Brick=GUID-or-FQN-of-Brick
```

OPTIONS **Brick**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Brick that was serviced.

EXAMPLES Run the following command to resume operations in your system after you replaced the Brick3 Brick.

```
pdscli request PerformResumeBrick @input
```

where input.req contains the following name/value pairs:

```
Request=PerformResumeBrick
Brick=/Company-XYZ-data/Brick3
```

See the PerformResumeBrickResponse.txt file for example response data.

SEE ALSO PerformIdentifyBrick on p. 433

PerformAcceptAddBrick on p. 391

Ranges for Option Values on p. 57

Service Guide

NAME **PerformResumeBrickFRU**

DESCRIPTION Resumes operations in the Pillar Axiom storage system after the:

- Removal of a field replaceable unit (FRU) in a Brick hardware component.
- Addition of a new FRU to replace the old one.

SYNTAX

```
pdscli {request | submit} PerformResumeBrickFRU
   Brick=GUID-or-FQN-of-Brick
   FRUName={PowerSupplyFanModule | ESModule |
      DiskDrive | SpareDiskDrive | RAIDController
      Chassis}
   FRUNumber=integer
```

OPTIONS

Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Brick that was serviced.

FRUName

Identifies a field replaceable unit (FRU) on a Brick. Choose from:

- PowerSupplyFanModule, which specifies either of the two power supply/fan modules that are installed on a Brick.
- ESModule, which specifies the enclosure services (ES) module that is installed on a Brick.
- DiskDrive, which specifies any of the six disk drives that are installed on each Brick control unit (CU).
- SpareDiskDrive, which specifies the spare disk drive that is shared by the two Brick CUs.
- RAIDController, which specifies either of the two RAID controllers that are installed on a Brick.
- Chassis, which specifies an entire Brick CU.

FRUNumber [optional]

Identifies a field replaceable unit (FRU) on a hardware component that is installed on the Pillar Axiom system.

EXAMPLES Run the following command to resume operations in your system afteryou replaced one of the array disk drives on a Brick.

```
pdscli request PerformResumeBrickFRU @input
```

where input.req contains the following name/value pairs:

```
Request=PerformResumeBrickFRU
Brick=/Company-XYZ-data/Brick3
FRUName=DiskDrive
FRUNumber=42
```

See the PerformResumeBrickFRUResponse.txt file for example response data.

SEE ALSO

PerformIdentifyBrickFRU on p. 435

PerformPrepareReplaceBrickFRU on p. 461

Ranges for Option Values on p. 57

Service Guide

NAME **PerformResumeSlammer**

DESCRIPTION Resumes operations in the Pillar Axiom storage system after the:

- Removal of a Slammer chassis.
- Addition of a new Slammer to replace the old one.

SYNTAX

```
pdscli {request | submit} PerformResumeSlammer
   Slammer=GUID-or-FQN-of-Slammer
   ControlUnitNumber={0 | 1}
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Slammer that was serviced.

ControlUnitNumber [optional]

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

EXAMPLES Run the following command to resume operations in your system after you replaced control unit (CU) 1 on the Slammer1 Slammer.

```
pdscli request PerformResumeSlammer @input
```

where input.req contains the following name/value pairs:

```
Request=PerformResumeSlammer
Slammer=/Company-XZY-data/Slammer1
ControlUnitNumber=1
```

See the PerformResumeSlammerResponse.txt file for example response data.

SEE ALSO PerformIdentifySlammer on p. 438

PerformAcceptAddSlammer on p. 392

Ranges for Option Values on p. 57

Service Guide

NAME **PerformResumeSlammerFRU**

DESCRIPTION Resumes operations in the Pillar Axiom storage system after the:

- Removal of a field replaceable unit (FRU) in a Slammer hardware component.
- Addition of a new FRU to replace the old one.

SYNTAX

```
pdscli {request | submit} PerformResumeSlammerFRU
   Slammer=GUID-or-FQN-of-Slammer
   ControlUnitNumber={0 | 1}
   FRUName={PowerSupply | FanModule |
      Battery |
      Memory | Motherboard |
      NASNetworkInterfaceModule |
      SANNetworkInterfaceModule |
      PrivateInterconnectModule | SCSIController |
      Chassis}
   FRUNumber=integer
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Slammer that was serviced.

ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

FRUName

Identifies a field replaceable unit (FRU) on a Slammer. Choose from:

- PowerSupply, which specifies either of the two power supplies that are installed on each Slammer control unit (CU). Each Slammer contains four power supplies.
- FanModule, which specifies either of the two fan assemblies that are installed on each Slammer CU. Each fan assembly contains two fans (which are not individual FRUs).
- Battery, which specifies either of the two batteries that are installed on each Slammer CU.
- Memory, which specifies memory modules that reside on the motherboard that is installed on each Slammer CU.

- Motherboard, which specifies the motherboard that is installed on each Slammer CU.
- NASNetworkInterfaceModule, which specifies either of the two network interface modules that are installed on a NAS Slammer. Both Slammer control units (CUs) must contain NAS or SAN network interface modules.
- SANNetworkInterfaceModule, which specifies either of the two network interface modules that are installed on a SAN Slammer.
- PrivateInterconnectModule, which specifies either of the two private interconnect modules that are installed on a Slammer.
- SCSIController, which specifies the PCI card that resides in the network interface module that is installed on each Slammer CU.
- Chassis, which specifies an entire Slammer CU.

FRUNumber [optional]

Identifies a field replaceable unit (FRU) on a hardware component that is installed on the Pillar Axiom system.

EXAMPLES

Run the following to resume operations in your system after you replaced the network interface module on the Slammer1 Slammer.

```
pdscli request PerformResumeSlammerFRU @input
```

where input.req contains the following name/value pairs:

```
Request=PerformResumeSlammerFRU
Slammer=/Company-XZY-data/Slammer1
ControlUnitNumber=1
FRUName=NASNetworkInterfaceModule
FRUNumber=42
```

See the PerformResumeSlammerFRUResponse.txt file for example response data.

SEE ALSO

PerformIdentifySlammerFRU on p. 440

PerformPrepareReplaceSlammerFRU on p. 463

Ranges for Option Values on p. 57

Service Guide

NAME **PerformSANiSCSIPing**

DESCRIPTION Sends an echco request to iSCSI data storage devices through the TCP/IP network, including iSCSI initiators (such as File Servers) and iSCSI targets (such as disk arrays and tape subsystems).

SYNTAX

```
pdscli {request | submit} PerformSANiSCSIPing
   Slammer=GUID-or-FQN-of-Slammer
   ControlUnitNumber={0 | 1}
   iSCSIPortName {PORT1, PORT0}
   IPAddress [1-255].[0-255].[0-255].[0-255]
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer that was serviced.

iControlUnitNumber [optional]

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs.

iSCSIPortName

Identifies the name (0 or 1) of the iSCSI port device.

IPAddress

Identifies the IP address that is assigned to the iSCSI port device.

EXAMPLES Run the following command to ping iSCSI devices.

```
pdscli request PerformSANiSCSIPing @input
```

where input.req contains the following name/value pairs:

```
Request=PerformSANiSCSIPing
Slammer=/Company-XZY-data/Slammer1
ControlUnitNumber=1
iSCSIPortName=Port0
IPAddress=172.20.67.0
```

See the PerformSANiSCSIPingResponse.txt file for example response data.

SEE ALSO ModifyiSCSIPortDetails on p. 343

Ranges for Option Values on p. 57

NAME **PerformSendSystemInformationToCallHomeServer**

DESCRIPTION Sends system information using the Call-Home connection between a Pillar Axiom storage system and the Pillar World Wide Customer Support Center.

Call-Home is a Pillar Axiom feature that, when enabled, allows the system to notify the Pillar World Wide Customer Support Center of critical issues specific to a Pillar Axiom system. No customer data is transmitted. Call-Home transfers files over the Internet using one of the following user-selected methods:

- SCP: Uses secure copy (SCP) with 1024-bit encryption and secure keys.
- HTTPS: Sends files either directly to the Pillar World Wide Customer Support Center or through a proxy server for security purposes. Can also be used when the Pillar Axiom system does not have direct access to the Internet.

When a component operates in degraded mode or fails, the system automatically performs failover actions. Although acomponent failure does not cause downtime, manual intervention is sometimes required to repair or replace the failed component. The system sends a Call-Home message to initiate the repair or replacement process.

SYNTAX

```
pdscli {request | submit}
PerformSendSystemInformationToCallHomeServer
  SystemInformationFile=name-of-system-information-file
```

OPTIONS None.

EXAMPLES Run the following command to send system information to the Pillar World Wide Customer Support Center using the Call-Home connection.

```
pdscli request PerformSendSystemInformationToCallHomeServer
```

See the PerformSendSystemInformationToCallHomeServerResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **PerformShutdown**

DESCRIPTION Places the Pillar Axiom storage system into a shutdown state. This option does not power off the system. Use this option in preparation for a power-off or power cycle, or for maintenance or recovery.

Important! If you need to power off the system for more than 48 hours, remove the Slammer batteries.

SYNTAX

```
pdscli {request | submit} PerformShutdown
```

OPTIONS None.

EXAMPLES Run the following command to shut down your system.

```
pdscli request PerformShutdown
```

See the PerformShutdownResponse.txt file for example response data.

SEE ALSO PerformRestart on p. 484

Ranges for Option Values on p. 57

NAME **PerformSlammerCommand**

DESCRIPTION Sends specified command to the Slammer control unit. You may use this request to transport a “pass-through” command to a Slammer node.

SYNTAX

```
pdscli {request | submit} PerformSlammerCommand
   Slammer=GUID-or-FQN-of-Slammer
   ControlUnitNumber={0 | 1}
   TimeoutSeconds=integer
   EnvironmentVariableList.
      EnvironmentVariable.Name=string
   EnvironmentVariableList.
      EnvironmentVariable.Value=integer
   Command=Command-name-and-arguments
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer.

ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs. The CUs are identified by 0 (the top CU) and 1 (the bottom CU).

TimeoutSeconds

Enter an integer that specifies an inactivity time limit, in seconds, after which the connection is terminated.

EnvironmentVariableList.EnvironmentVariable.Name [optional]

Identifies the name that is assigned to an environmental variable.

EnvironmentVariableList.EnvironmentVariable.Value [optional]

Identifies the value assigned to the environmental variable.

Command

Identifies a command to perform in order to resolve a connectivity issue between the specified Pillar Axiom Slammer and the customer network. Enter the command and command-line arguments:

- required arguments
- [optional arguments]

A File Server configuration contains virtual interfaces (VIFs). The name of a File Server is internally mapped to a socket identifier (SOCK) and a virtual server identifier (VSID). The environment variables SOCK and VSID provide values to various Slammer commands which then use those values to identify the target File Server.

The syntax of the SOCK and VSID environment variables is defined as follows:

Table 29 SOCK and VSID environment variables

Variable	Description
SOCK=*/vserverN*	Provides a prefix that the system uses to address an internal structure associated with a particular File Server. When this variable is not set, the system addresses a more generic structure within the internal network that interconnects the Pilot and the Slammers (this network is called the private management interface, or PMI). Most commonly, this variable is set to a particular value in order to run a command against the data interface.
VSID=*virtualServerID*	Identifies by number (*virtualServerID*) a particular File Server.

Note: For a given File Server, the value for N and the value for virtualServerID are the same value.

Tip: If you have one (and have had only one) File Server, the value of SOCK would be */vserver0* and the value for VSID would be 0. Otherwise, you can execute `ifconfig` repetitively using various settings for SOCK and checking each returned IP address against the address configured for the control unit to determine the correct value for SOCK. Although the maximum number of supported File Servers is eight at any given time, the maximum value for SOCK is */vserver255* and that for VSID is 255.

Table 30 Resolve connectivity trouble commands

Command	Arguments	Description
`ping`	-c *count ipaddr* • -c count = Replace count with an unassigned number, which is the number of packets to send. • *ipaddr* = IP address of a host machine. Environment variables (optional): SOCK=*node_ID*	Sends an echo request to a specified host. Default sends the echo request from the Private Management Interface (PMI) unless the SOCK environment variable is set. If used, this variable must be set to the virtual interface from where you want the echo request to be sent. Example: The following command requests an echo from the host whose IP is 198.168.1.2 by sending five packets to that address. `ping -c 5 198.168.1.2`

`route`	`route show`
	Environment variables (optional):
	SOCK=*node_ID*
		Displays the route table.
		Default displays the route table of the PMI unless the SOCK environment variable is set. If used, this variable must be set to the virtual interface of the route table that you want to display. The Pilot and the Slammer communicate to one another over the PMI.
		Example:
		The following command displays the route table associated with virtual server 0.
		`route show`
		`SOCK=/vserver0`

`traceroute`	[`-s` *ipaddr*] [`-r`] [`-v`] *host* • [`-s` *ipaddr*] = Use the given IP address as the source address of the traceroute probe packet. • [`-r`] = Bypass the normal routing tables and send directly to the attached host. • [`-v`] = Verbose mode. • *host* = The target host. Environment variables (optional): SOCK=*node_ID*	Traces the route that an IP packet takes to reach a specified host. Default sends the probe packet from the PMI interface unless the SOCK environment variable is set. If used, this variable must be set to the virtual interface from where you want the traceroute probe to come. Example: The following command displays all route information between the PMI and the host whose IP address is 10.20.5.29. `traceroute -v 10.20.5.29`

Table 30 Resolve connectivity trouble commands (continued)

Command	Arguments	Description

`arp`	`-a`
	-a = Display current ARP entries.
	Environment variables (optional):
	SOCK=*node_ID*
		Displays the IP-to-physical address translation tables used by the Address Resolution Protocol (ARP).
		Default displays the ARP table of the PMI interface, unless the SOCK environment variable is set.
		Example:
		The following command displays the ARP table located in the Pilot.
		`arp -a`

`ifconfig`	Environment variables (optional):
	SOCK=*node_ID*
		Displays interfaces and related information.
		Default displays interface information for the PMI, unless the SOCK environment variable is set.
		Example:
		`ifconfig`

`wbinfo`	[-t] [-m] [-I ipaddr]
	[-N name]
	• -t = Verifies that the workstation trust account that was created when the CIFS server was added to the Windows domain is working and that the shared secret is good.
	• -m = Lists trusted domains.
	• -Iipaddr = Converts an IP address to a NetBIOS name (WINS).
	• -N name = Converts a NetBIOS name to an IP (WINS).
	Environment variables (required):
	SOCK=*VSID*
		Queries information from the CIFS windbind daemon. By default, this command displays the help information.
		Note: Only the parameters listed in the Syntax column are supported for this command.
		Example:
		The following command requests the CIFS server to confirm that the status of the joined domain is Yes.
		`wbinfo -t`
		`VSID=0 SOCK=/vserver0`

Table 30 Resolve connectivity trouble commands (continued)

Command	Arguments	Description

`nmblookup`	[-U *server]* [-R] *hostName*
	• -U *server* = Specifies IP address or name of the WINS server. Pillar recommends using IP address to be consistent with the value configured in the File Server definition.
	• -R = Causes the WINS server to return all names stored on the server.
	• *hostName* = Name of a NetBIOS host.
	Environment variables (required):
	• SOCK=*/vserverN*
	• VSID=*serverID*
		Looks up NetBIOS names on a WINS server and maps them to IP addresses.
		This command requires that both the SOCK and VSID environment variables be set.
		Example 1:
		The following command requests the WINS server at 172.10.1.20 to resolve the cifs01 CIFS server to an IP address.
		`nmblookup -U 172.10.1.20`
		`-R cifs01`
		`VSID=0 SOCK=/vserver0`
		Example 2:
		The following command requests the WINS server at 172.10.1.20 to provide the IP addresses of the domain controllers serving the domain01 NTLM domain.
		`nmblookup -U 172.10.1.20`
		`-R domain01#1c`
		`VSID=0 SOCK=/vserver0`
		Note: The two hexadecimal digits (1c) following the # character specify the NetBIOS name type as Domain Controllers, which conforms to the NetBIOS suffix convention.

`nslookup`	*hostName* [*serverName*]
	• *hostName* = Display information about the specified host.
	• *serverName* = Specifies the name of the DNS server.
	Environment variables (optional):
	SOCK=*node_ID*
		Queries domain name server (DNS).
		Default sends the domain name request over the PMI interface unless the SOCK environment variable is set. If used, this variable must be set to the virtual interface from where you want the query to be sent.
		Example:
		The following command looks up the IP address of ca-lab.eng using the DNS at 10.20.0.10. In this example, the request is sent from the NAS Slammer port that is associated with virtual server 0.
		`nslookup ca-lab.eng`
		`10.20.0.10`

Table 30 Resolve connectivity trouble commands (continued)

Command	Arguments	Description
`perf`	[`-c`][-l *numThreads*] • -c = Rests the counters after returning statistics. • -l*numThreads* = Displays the statistics for the specified number of threads that consume the most CPU cycles. Environment variables: None	Checks CPU utilization of the Slammer. This command returns the CPU statistics for the idle process, the kernel time, and all process IDs that are running. When you use the -c option, the next invocation of `perf` will contain data accumulated from that point. Example: The following command displays the utilization statistics for all threads running on the selected Slammer and then clears the counters for that Slammer. `perf -c`

`netstat`	[-s] [-r] [-pip] [-pudp] [-ptcp]
	• -s = Request statistics instead of connections.
	• -r = Displays the routing table.
	• -pip = Displays IP level statistics. Good for finding issues with fragments for NFS over UDP.
	• -pudp = Displays UDP level statistics.
	• -ptcp = Displays TCP level statistics.
	Environment variables (required):
	SOCK=*node_ID*
		Shows networking statistics for a NAS Slammer control unit. If no options are specified, *netstat* returns information about the listening sockets and established connections.
		Example:
		The following command requests the IP packet statistics that are associated with virtual server 0.
		`netstat -pip`
		`SOCK=/vserver0`

EXAMPLES

Run the following Slammer command.

```
pdscli request PerformSlammerCommand @input
```

where input.req contains the following name/value pairs:

```
Request=PerformSlammerCommand
Slammer=/XZY-data/Slammer1
ControlUnitNumber=1
TimeoutSeconds=145
Command=ifconfig
```

See the PerformSlammerCommandResponse.txt file for example response data.

SEE ALSO GetSlammerCommandOutput on p. 292

Ranges for Option Values on p. 57

NAME **PerformSlammerControlUnitDiagnostic**

DESCRIPTION Runs diagnostics for a specified Slammer control unit.

Important! Contact the Pillar World Wide Support Center before running this request.

SYNTAX

```
pdscli {request | submit} PerformSlammerControlUnitDiagnostic
     Slammer=GUID-or-FQN-of-Slammer
     ControlUnitNumber={0 | 1}
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the Slammer on which a FRU will be replaced when the system is ready.

ControlUnitNumber

Identifies a specific control unit (CU) in a Slammer. Each Slammer contains two CUs. The CUs are identified by 0 (the top CU) and 1 (the bottom CU).

EXAMPLES Run the following command to perform diagnostics on control unit (CU) 1 on Slammer1.

```
pdscli request PerformSlammerControlUnitDiagnostic @input
```

where input.req contains the following name/value pairs:

```
Request=PerformSlammerControlUnitDiagnostic
Slammer=/Company-XZY-data/Slammer1
ControlUnitNumber=1
```

See the PerformSlammerControlUnitDiagnosticResponse.
txt file for example response data.

SEE ALSO GetSlammerStatistics on p. 296

Ranges for Option Values on p. 57

NAME **PerformTapeRobotDiscovery**

DESCRIPTION Discovers tape devices that are attached to the Pillar Axiom storage system so that administrators can back up data that is stored on the system.

SYNTAX

```
pdscli {request | submit} PerformTapeRobotDiscovery
```

OPTIONS None.

EXAMPLES Run the following command to make a tape device known to your system.

```
pdscli request PerformTapeRobotDiscovery
```

See the PerformTapeRobotDiscoveryResponse.txt file for example response data.

SEE ALSO GetAllTapeRobots on p. 234

Ranges for Option Values on p. 57

NAME **PerformTestCallHome**

DESCRIPTION Tests the Call-Home connection between a Pillar Axiom storage system and the Pillar World Wide Customer Support Center.

SYNTAX

```
pdscli {request | submit} PerformTestCallHome
```

OPTIONS None.

EXAMPLES Run the following command to test your Call-Home connection.

```
pdscli request PerformTestCallHome
```

See the PerformTestCallHomeResponse.txt file for example response data.

SEE ALSO PerformPeriodicCallHome on p. 460

Ranges for Option Values on p. 57

NAME

PerformTestEmail

DESCRIPTION

Tests the connection between a Pillar Axiom storage system and the email server that is defined in the system's Global Settings.

SYNTAX

```
pdscli {request | submit} PerformTestEmail
   EmailAddress[1-4]=username@domain.ext
```

OPTIONS

EmailAddress[1-4]

Identifies the recipient's email address. The email server to which the Pillar Axiom system sends alerts must be able to send messages to this email address.

EXAMPLES

Run the following command to test whether your system can communicate with the email server that you specified in Global Settings.

```
pdscli request PerformTestEmail @input
```

where input.req contains the following name/value pairs:

```
Request=PerformTestEmail
EmailAddress1=xavier@xyz.com
```

See the PerformTestEmailResponse.txt file for example response data.

SEE ALSO

CreateNotificationConnection on p. 151

CreateNotificationDefinition on p. 152

Ranges for Option Values on p. 57

NAME **PerformUpdate**

DESCRIPTION Installs the specified firmware or software module.

SYNTAX

```
pdscli {request | submit} PerformUpdate
   UpdatePackage=/FQN-of-package
   UpdateComponentName=
      {"Pilot Software" | "Pilot OS" |
      "Slammer Software" | "Slammer PROM" |
      "Slammer Software AX600" | "Slammer PROM AX600" |
      "Brick Firmware" | Brick FC Firmware}
   PerformUpgradeDisruptively={true | false}
```

OPTIONS

UpdatePackage

Enter the fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

FQN format:

```
/Fully/Qualified/ObjectName
```

UpdateComponentName

Identifies the name of an updatable component. Choose between one and six of the following options.

Table 31 Updatable components

Name	Description
"Pilot Software"	Software that runs on the Pilot, such as the GUI interface and web server, online help, SNMP, and NDMP
"Pilot OS"	Operating system for the Pilot
"Slammer Software"	NAS or SAN software that runs on Pillar Axiom 300 and 500 Slammers
"Slammer Software AX600"	NAS or SAN software that runs on Pillar Axiom 600 Slammers
"Slammer PROM"	Programmable ROM (PROM), that includes BIOS and netboot code, for Pillar Axiom 300 and 500 NAS or SAN Slammers
"Slammer PROM AX600"	Programmable ROM (PROM), that includes BIOS and netboot code, for Pillar Axiom 600 Slammers
"Brick Firmware"	RAID firmware for SATA Bricks
"Brick FC Firmware"	FC firmware for Bricks

Table 31 Updatable components (continued)

Name	Description
“Brick SATA2 Firmware”	RAID firmware for version 2 SATA Bricks

PerformUpgradeDisruptively

If enabled, all protocol services are shut down at the time of the software update and are not restored until the update is complete. During the software update, the system shuts down and restarts. Depending on the size of the configuration, this disruption could exceed thirty minutes. The normal and recommended use of this option is to leave it disabled.

EXAMPLES

Run the following command to update the Pilot software in your system.

```
pdscli request PerformUpdate @input
```

where input.req contains the following name/value pairs:

```
Request=PerformUpdate
UpdatePackage=/Staged
UpdateComponentName="Pilot Software"
PerformUpgradeDisruptively=false
```

See the PerformUpdateResponse.txt file for example response data.

SEE ALSO

CreateJob on p. 127

PerformUpdatePackageStaging on p. 511

PerformUpdateValidation on p. 513

Ranges for Option Values on p. 57

NAME **PerformUpdateDiskDrive**

DESCRIPTION Updates firmware on the Brick disk drives.

SYNTAX

```
pdscli {request | submit} PerformUpdateDiskDrive
   UpdatePackage=/FQN-of-package
   UpdateComponentName=Name-of-updatable-component
```

OPTIONS

UpdatePackage

Enter the fully qualified name (FQN), which starts with a leading slash (/), contains a parent object’s name if needed to establish uniqueness of the object, and ends with the object’s name.

FQN format:

```
/Fully/Qualified/ObjectName
```

UpdateComponentName

Identifies the name of an updatable component.

EXAMPLES Run the following command to update the disk drives in your system.

```
pdscli request PerformUpdateDiskDrive @input
```

where input.req contains the following name/value pairs:

```
Request=PerformUpdateDiskDrive
UpdatePackage=/Staged
UpdateComponentName="Brick Disk Drive Firmware 2052-00003"
```

See the PerformUpdateDiskDriveResponse.txt file for example response data.

SEE ALSO PerformUpdateDiskDriveValidation on p. 510

PerformUpdate on p. 507

PerformUpdatePackageStaging on p. 511

PerformUpdateValidation on p. 513

Ranges for Option Values on p. 57

NAME **PerformUpdateDiskDriveValidation**

DESCRIPTION Validates the firmware update file for Brick disk drives.

SYNTAX

```
pdscli {request | submit} PerformUpdateDiskDriveValidation
   UpdatePackage=/FQN-of-package
   UpdateComponentName=Name-of-updatable-component
```

OPTIONS

UpdatePackage

Enter the fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

FQN format:

```
/Fully/Qualified/ObjectName
```

UpdateComponentName

Identifies the name of an updatable component.

EXAMPLES

Run the following command to verify that new firmware is compatible with other software packages.

```
pdscli request PerformUpdateDiskDriveValidation @input
```

where input.req contains the following name/value pairs:

```
Request=PerformUpdateDiskDriveValidation
UpdatePackage=/Staged
UpdateComponentName="Brick Disk Drive Firmware 2052-00003"
```

See the PerformUpdateDiskDriveValidationResponse.txt file for example response data.

SEE ALSO

PerformUpdateDiskDrive on p. 509

PerformUpdate on p. 507

PerformUpdatePackageStaging on p. 511

PerformUpdateValidation on p. 513

Ranges for Option Values on p. 57

NAME

PerformUpdatePackageStaging

DESCRIPTION

Uploads and expands the specified update file in preparation for a software or firmware update.

SYNTAX

```
pdscli {request | submit} PerformUpdatePackageStaging
   UpdatePackageFilename=.tar.gz-file-name
   FileContents="&@filename;"
```

Tip: You can use “ignoreme” as the package file name, and then specify the path to the package. The quotes, the ampersand (&), the @ sign, and the ending semicolon are mandatory characters.

OPTIONS

UpdatePackageFilename [optional]

Identifies the name of an update file.

- The staging process uploads the file to the Pilot and expands the compressed .tar.gz file.
- The validation process checks the compatibility of the firmware and software in the update file against previously installed firmware and software.
- The update process installs the uncompressed package on the Pillar Axiom storage system.

Select a single update file for the given operation.

FileContents [optional]

Identifies that the Pillar Axiom system should Base64 encode the contents of the specified file and upload the encoded file. The `&`, `@` and `;` characters that surround the file name tell the CLI to encode and upload the file.

EXAMPLES

Run the following command to stage an update file for a software update.

```
pdscli request PerformUpdatePackageStaging @input
```

where input.req contains the following name/value pairs:

```
Request=PerformUpdateFileStaging
UpdatePackageFilename="ignoreme"
FileContents="&@/myupdate/packages/axiom_package_1U.tar.gz.020304-
000040;"
```

See the PerformUpdatePackageStagingResponse.txt file for example response data.

SEE ALSO

PerformUpdate on p. 507

PerformUpdateValidation on p. 513

Ranges for Option Values on p. 57

NAME **PerformUpdateSystemSerialNumber**

DESCRIPTION Deletes **all** user data and configuration details on the Pillar Axiom storage system. To perform this request, you must get an encrypted file with a new system serial number (SSN) from the Pillar World Wide Customer Support Center.

Note: This request clears the system configuration, including all groups and LUNs. It performs a system reset.

SYNTAX

```
pdscli {request | submit}
  PerformUpdateSystemSerialNumber
    SystemSerialNumberFileName=string
    SystemSerialNumberFileContents=
      "&@filename;"
    EncryptedSystemSerialNumber=string
```

OPTIONS

SystemSerialNumberFileName

Identifies the name of file that you receive from Technical Support, which contains the encrypted system serial number.

SystemSerialNumberFileContents

Identifies that the Pillar Axiom system should Base64 encode the contents of the specified file and upload the encoded file. The `&`, `@` and `;` characters that surround the file name tell the CLI to encode and upload the file.

EncryptedSystemSerialNumber

Identifies the encrypted system serial number (SSN) that is assigned to the Pillar Axiom storage system.

EXAMPLES Run the following command to delete all of the data and configuration details on your system and reset the system serial number (SSN).

```
pdscli request PerformUpdateSystemSerialNumber
  @input
```

where input.req contains the following name/value pairs:

```
Request=PerformUpdateSystemSerialNumber
SystemSerialNumberFileName=Company-XYZ-ssn
SystemSerialNumberFileContents="&@XYZ-ssn;"
EncryptedSystemSerialNumber=A000001BCD
```

See the PerformUpdateSystemSerialNumber Response.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **PerformUpdateValidation**

DESCRIPTION Validates the specified update file for compatibility with installed software and firmware packages.

SYNTAX

```
pdscli {request | submit} PerformUpdateValidation
   UpdatePackage=.tar.gz-file-name
   UpdateComponentName=
      {"Pilot Software" | "Pilot OS" |
      "Slammer Software" | "Slammer PROM" |
      "Slammer Software AX600" | "Slammer PROM AX600" |
      "Brick Firmware" | Brick FC Firmware}
   PerformUpgradeDisruptively={true | false}
```

OPTIONS

UpdatePackage

Identifies the name of an update file.

- The staging process uploads the file to the Pilot and expands the compressed .tar.gz file.
- The validation process checks the compatibility of the firmware and software in the update file against previously installed firmware and software.
- The update process installs the uncompressed package on the Pillar Axiom storage system.

Select a single update file for the given operation.

UpdateComponentName

Identifies the name of an updatable component. Choose between one and six of the following options.

Table 32 Updatable components

Name	Description
“Pilot Software”	Software that runs on the Pilot, such as the GUI interface and web server, online help, SNMP, and NDMP
“Pilot OS”	Operating system for the Pilot
“Slammer Software”	NAS or SAN software that runs on Pillar Axiom 300 and 500 Slammers
“Slammer Software AX600”	NAS or SAN software that runs on Pillar Axiom 600 Slammers
“Slammer PROM”	Programmable ROM (PROM), that includes BIOS and netboot code, for Pillar Axiom 300 and 500 NAS or SAN Slammers

Table 32 Updatable components (continued)

Name	Description
"Slammer PROM AX600"	Programmable ROM (PROM), that includes BIOS and netboot code, for Pillar Axiom 600 Slammers
"Brick Firmware"	RAID firmware for SATA Bricks
"Brick FC Firmware"	FC firmware for Bricks
"Brick SATA2 Firmware"	RAID firmware for version 2 SATA Bricks

PerformUpgradeDisruptively

This flag is used to allow the administrator to specify whether an update for validation is to be done disruptively or non-disruptively.

EXAMPLES

Run the following command to verify that new Pilot software is compatible with other software packages.

```
pdscli request PerformUpdateValidation @input
```

where input.req contains the following name/value pairs:

```
Request=PerformUpdateValidation
UpdatePackage=mgmt_upgrade0003.tar.gz
UpdateComponentName="Pilot Software"
PerformUpgradeDisruptively=false
```

See the PerformUpdateValidationResponse.txt file for example response data.

SEE ALSO

PerformUpdate on p. 507

Ranges for Option Values on p. 57

NAME **PerformVerifyDataConsistency**

DESCRIPTION Verifies the integrity of user data that is stored on the specified Brick.

SYNTAX

```
pdscli {request | submit}
   PerformVerifyDataConsistency
      Brick=GUID-or-FQN-of-Brick
      Priority={High | Low}
```

OPTIONS

Brick

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.

Enter the GUID or FQN of the Brick to verify.

Priority

Identifies the priority to assign to the verification check. Choose from:

- High to permit the verification check to affect performance by up to 30%.
- Low to permit the verification check to affect performance by up to 10%.

EXAMPLES Run the following command to verify data consistency on Brick1 that is configured on your system.

```
pdscli request PerformVerifyDataConsistency @input
```

where input.req contains the following name/value pairs:

```
Request=PerformVerifyDataConsistency
Brick=/Brick1
Priority=Low
```

See the PerformVerifyDataConsistencyResponse.txt file for example response data.

SEE ALSO GetDataConsistencyDetails on p. 249

Ranges for Option Values on p. 57

NAME **PerformVerifyStorageRedundancy**

DESCRIPTION Verifies the redundancy settings on the specified volume. All of the options are mutually exclusive. Use the option of the volume type for which you are verifying.

SYNTAX

```
pdscli {request | submit}
   PerformVerifyStorageRedundancy
      FileSystem=GUID-or-FQN-of-filesystem
      CloneFS=GUID-or-FQN-of-cloneFS
      LUN=GUID-or-FQN-of-LUN
      SnapLUN=GUID-or-FQN-of-cloneLUN
```

OPTIONS

FileSystem [Optional]

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem to verify.

CloneFS [Optional]

Identifies the name that the Pillar Axiom system assigns to a Clone FS.

LUN [Optional]

Identifies the name that is assigned to a LUN for administrative purposes. LUN names must be unique across the Pillar Axiom system, not just within its associated volume group.

SnapLUN [Optional]

Identifies the name that the Pillar Axiom system assigns to a Clone LUN (formally referred to as Snap LUN).

EXAMPLES Run the following command to verify the redundancy on the engineering organization's filesystem.

```
pdscli request PerformVerifyStorageRedundancy
   @input
```

where input.req contains the following name/value pairs:

```
Request=PerformVerifyStorageRedundancy
FileSystem=/Company-XYZ-FS1/EngAtl
```

See the PerformVerifyStorageRedundancyResponse.txt file for example response data.

SEE ALSO

GetStorageRedundancyDetails on p. 301

Ranges for Option Values on p. 57

NAME **PrepareSnapLUN**

DESCRIPTION Prepares a Snap LUN, which is defined as:

> A point-in-time, read-write copy of a LUN that you can immediately use. Snap LUNs requires no cache and minimal disk storage space.
>
> Snap LUNs point to the original data, and have the same QoS parameters as the source LUN.
>
> Note: Formerly called a Snap LUN, the GUI now refers to this term as Clone LUN, whereas Snap LUN is still used in the command line.

After the Snap LUN is prepared, it can be synched using the SyncSnapLUN command.

SYNTAX

```
pdscli {request | submit} PrepareSnapLUN
   Source=GUID-or-FQN-of-SnapLUN
   SnapLUNName=LUN-name
   LUNNumber=integer-0-through-255
   Mapped={true | false}
   Active={true | false}
   EnableiSCSIAccess={true | false}
   EnableFibreChannelAccess={true | false}
```

OPTIONS

SnapLUN

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the LUN you wish to modify.

SnapLUNName [optional]

Identifies the name that is assigned to a LUN for administrative purposes. LUN names must be unique across the Pillar Axiom system, not just within its associated volume group.

Name is a required parameter only if you want to change the FQN LUN's user-visible name.

LUNNumber [optional]

Identifies the unique number that is assigned to a LUN and can be accessed by all SAN hosts if the LUN is not mapped. If the LUN is mapped, the LUN number must be unique to the mapped SAN host.

Mapped [optional]

Identifies whether all hosts or only selected hosts may access this LUN.

- Mapped allows only designated hosts to access this LUN.
- Unmapped allows all hosts to access this LUN using the LUN Number.

Note: If you choose Mapped, do not assign a LUNNumber.

Active [optional]

Identifies whether the Clone LUN is available to the host (active) or is not seen by the host (inactive) when it is created. Inactive Clone LUNs can be changed to active; however, once the Clone LUN is set to active, you cannot change it to inactive.

EnableiSCSIAccess

Identifies that the iSCSI protocol specification is used when accessing the newly created SAN LUN.

Note: You must specify iSCSI, Fibre Channel, or both protocols.

EnableFibreChannelAccess

Identifies that the Fibre Channel (FC) protocol specification is used when accessing the newly created SAN LUN.

Note: You must specify iSCSI, Fibre Channel, or both protocols.

EXAMPLES

Run the following command to prepare SLUN_01 for the Engineering organization.

```
pdscli request PrepareSnapLUN @input
```

```
where input.req contains the following name/value pairs:
Request=PrepareSnapLUN
Source=/SLUN_01
SnapLUNName=EngLUN
LUNNumber=0
Mapped=false
Active=false
EnableiSCSIAccess=true
EnableFibreChannelAccess=false
```

See the PrepareSnapLUNResponse.txt file for example response data.

SEE ALSO

DeleteSnapLUN on p. 191

GetAllSnapLUNs on p. 233

GetAllSnapLUNHierarchies on p. 232

GetSnapLUNDetails on p. 297

SyncSnapLUN (p. 529)

Ranges for Option Values on p. 57

NAME

ReenableAllRAIDControllers

DESCRIPTION

Important! Reenables RAID controllers.

SYNTAX

```
pdscli {request | submit} ReenableAllRAIDControllers
```

OPTIONS

None.

EXAMPLES

Run the following request:

```
pdscli request ReenableAllRAIDControllers @input
```

where input.req contains the following name/value pairs:

```
Request = ReenableAllRAIDControllers
```

See the RenenableAllRAIDControllersResponse.txt file for example response data.

SEE ALSO

ReenableAllSlammers on p. 522

ReenableCUBoot on p. 523

NAME	**ReenableAllSlammers**

DESCRIPTION Re-enables all Slammers instead of just one.

SYNTAX

```
pdscli {request | submit} ReenableAllSlammers
```

OPTIONS None.

EXAMPLES Run the following command to re-enable all Slammers.

```
pdscli request ReenableAllSlammers @input
```

where input.req contains the following name/value pairs:

```
Request = ReenableAllSlammers
```

See the ReenableAllSlammersResponse.txt file for example response data.

SEE ALSO ReenableAllRAIDControllers on p. 521

ReenableCUBoot on p. 523

NAME **ReenableCUBoot**

DESCRIPTION This request only allows the control unit (CU) to boot. It does not send any signals to the Slammer CU. Typically, use the request only to recover a specific Slammer CU when there are multiple disabled CUs and you don't want to recover all of them using the ReenableAllSlammers request.

SYNTAX

```
pdscli {request | submit} ReenableCUBoot
   Slammer=GUID-or-FQN-of-Slammer
   ControlUnitNumber={0-for-automatic-assignment | integer}
```

OPTIONS

Slammer

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (0 through 9 and a through f) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name. Field types support UTFString.
  FQN Format:
  `/Fully/Qualified/ObjectName`

ControlUnitNumber [optional]

Specifies a specific control unit (CU) on the Slammer.

EXAMPLES Run the following request:

```
pdscli request ReenableCUBoot @input
```

where input.req contains the following name/value pairs:

```
Request = ReenableCUBoot
Slammer=Slammer1
```

See the RenableCUBootResponse.txt file for example response data.

SEE ALSO ReenableAllSlammers on p. 522

ReenableAllRAIDControllers on p. 521

NAME

Relogin

DESCRIPTION

Restarts a command line interface (CLI) session for the specified Pillar Axiom administrator or NDMP account. You may have to perform this request after the initial session times out. You can avoid the system timing out by increasing the session timeout value.

SYNTAX

```
pdscli {request | submit} Relogin
   Username=string
   Password=string
```

OPTIONS

Username

Enter a valid user name that is defined for a Pillar Axiom administrator account.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

Password

Enter a valid password that is defined for the specified user name.

Note: If you specify a value, it will override values you may have saved in configuration files or environment variables.

EXAMPLES

Run the following command to restart a management session that timed out.

```
pdscli request Relogin @input
```

where input.req contains the following name/value pairs:

```
Request=Relogin
Username=xavier
Password=s7Z12W79
```

See the ReloginResponse.txt file for example response data.

SEE ALSO

Login on p. 309

Logout on p. 310

Ranges for Option Values on p. 57

NAME **ResetCallHomeToDefaultSettings**

DESCRIPTION Restores Call-Home feature functionality to the default settings. The Call-Home feature:

Notifies the Pillar World Wide Customer Support Center about issues in the Pillar Axiom system. When a component operates in degraded mode or fails, the system automatically performs failover actions. Although acomponent failure does not cause downtime, manual intervention is sometimes required to repair or replace the failed component. The system sends a Call-Home message to initiate the repair or replacement process.

SYNTAX

```
pdscli {request | submit} ResetCallHomeToDefaultSettings
```

OPTIONS None

EXAMPLES Run the following command to reset the Call-Home functionality to the default settings.

```
pdscli request ResetCallHomeToDefaultSettings @input
```

See the ResetCallhomeToDefaultSettingsResponse.txt file for example response data.

SEE ALSO Ranges for Option Values on p. 57

NAME **RestoreFileSystemFromCloneFS**

DESCRIPTION Restores a filesystem using the content from a Clone FS, if the capacity of the Clone FS is less than or equal to that of the filesystem. If the clone capacity is greater than that of the filesystem, the system rejects the request.

As a result of the restore process:

- The snapshot schedule for the restored volume is erased. A new snapshot schedule for the restored filesystem will need to be re-created.
- The creation data for a SecureWORMfs is reset to the time of the restore.
- The filesystem size remains the same as before the restore.
- Quotas from the Clone FS are restored to the filesystem.
- The exports and shares belong to the source filesystem and will be unchanged by the restore from the Clone FS.

SYNTAX

```
pdscli {request | submit} RestoreFileSystemFromCloneFS
```

OPTIONS None

EXAMPLES Run the following command to restore a filesystem from a Clone FS.

```
pdscli request RestoreFileSystemFromCloneFS @input
```

See the RestoreFileSystemFromCloneFSResponse.txt file for example response data.

SEE ALSO CreateCloneFS (p. 97)

CreateFileSystem (p. 110)

DeleteCloneFS (p. 172)

DeleteFileSystem (p. 176)

GetAllCloneFSHierarchies (p. 207)

GetAllCloneFSs (p. 208)

GetAllFileSystems (p. 214)

GetCloneFSDetails (p. 245)

GetFileSystemDetails (p. 255)

ModifyCloneFS (p. 317)

ModifyFileSystem (p. 330)

NAME **RestoreLUNFromCloneLUN**

DESCRIPTION Restores a LUN using the content from a Clone LUN, if the capacity of the Clone LUN is less than or equal to that of the LUN. If the clone capacity is greater than that of the LUN, the system rejects the request.

LUN parameters remain unchanged by the LUN restore process, including:

- LUN size
- LUN mappings
- Masking
- LUN numbering
- Enabled protocols
- QoS

SYNTAX

```
pdscli {request | submit} RestoreLUNFromCloneLUN
```

OPTIONS None

EXAMPLES Run the following command to restore a filesystem from a Clone FS.

```
pdscli request RestoreLUNFromCloneLUN @input
```

See the RestoreLUNFromCloneLUNResponse.txt file for example response data.

SEE ALSO CreateLUN (p. 135)

CreateSnapLUN (p. 161)

DeleteLUN (p. 182)

DeleteSnapLUN (p. 191)

GetAllLUNs (p. 221)

GetAllSnapLUNHierarchies (p. 232)

GetAllSnapLUNs (p. 233)

GetLUNDetails (p. 269)

GetSnapLUNDetails (p. 297)

ModifyLUN (p. 351)

ModifySnapLUN (p. 381)

NAME **SNMPGetAllQuotas**

DESCRIPTION Displays a list of quotas, or capacity limits, that are defined for a specified filesystem.

Important! Attempting to retrieve definitions for more than 500 quotas for a specified filesystem is not recommended because of the potential negative impact to the Pilot's performance.

SYNTAX

```
pdscli {request | submit} GetAllQuotas
   ID=GUID-or-FQN-of-FileSystem
```

OPTIONS **ID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the filesystem for which you want to display a list of quotas.

EXAMPLES Run the following command to display a list of quotas for the EngAtl filesystem.

```
pdscli request GetAllQuotas @input
```

where input.req contains the following name/value pairs:

```
Request=GetAllQuotas
ID=/Company-XYZ-FS1/EngAtl
```

See the GetAllQuotasResponse.txt file for example response data.

NAME **SyncSnapLUN**

DESCRIPTION Synchronizes a Snap LUN, which is defined as:

Sy

> A point-in-time, read-write copy of a LUN that you can immediately use. Snap LUNs requires no cache and minimal disk storage space.
>
> Snap LUNs point to the original data, and have the same QoS parameters as the source LUN.

SYNTAX

```
pdscli {request | submit} SyncSnapLUN
  SnapLUNID=GUID-or-FQN-of-LUN
```

OPTIONS **SnapLUNID**

Identifies a specific Pillar Axiom object. Choose from:

- Globally unique ID (GUID), which is a 38-character string that starts with ID and ends with 36 lower-case hex characters (a through f and 0 through 9) and hyphens.
  Example GUID:
  `ID12345678-9abc-def0-1234-56789abcdef0`
- Fully qualified name (FQN), which starts with a leading slash (/), contains a parent object's name if needed to establish uniqueness of the object, and ends with the object's name.
  FQN Format:
  `/Fully/Qualified/ObjectName`

Enter the GUID or FQN of the LUN you wish to synchronize.

EXAMPLES Run the following command to respond to an AdministratorAction and delete pinned data from the EngAtl filesystem.

```
pdscli request SyncSnapLUN @input
```

where input.req contains the following name/value pairs:

```
Request=SyncSnapLUN
SnapLUNID=/Company-XYZ-FS1/EngAtl
```

See the SyncSnapLUNResponse.txt file for example response data.

See Also DeleteSnapLUN on p. 191

GetAllSnapLUNs on p. 233

GetSnapLUNDetails on p. 297

ModifySnapLUN on p. 381

PrepareSnapLUN on p. 518

APPENDIX A

CLI Return Status Codes

Introduction to Return Status Codes

As you use the `pdscli`, you may see the status codes that are documented in this section. Use appropriate utilities for your environment. For example:

- In Bash, get return status codes with:

  ```
  "$?"
  ```

- In Windows batch files, get return status codes with the ERRORLEVEL operator.

Return Status Codes

The CLI return status codes are shown in Table 33.

Note: A CLI request might return an error. To learn more about an error message, use the `pdscli help list-errors` command.

Table 33 CLI return status codes

Code	Meaning
0	Success
1	• Syntax errors on command line • Missing user name or password
2	Trace file errors

Table 33 CLI return status codes (continued)

Code	Meaning
3	Input errors: • File not found • Syntax error • Unknown field • Unknown request
4	Connection errors: • Unknown or missing host • Invalid port • Invalid timeout period
5	Host communication errors or timeout exceeded
6	Parse errors (Reply)
7	RequestStatus in the ReplyHeader is something other than SUCCEEDED

Index